

Unscharfe stochastische Optimierung und Anwendungen im Marketing

Von der Fakultät für Mathematik, Naturwissenschaften und Informatik
der Brandenburgischen Technischen Universität Cottbus

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Dr. rer. nat.)

genehmigte Dissertation

vorgelegt von

Diplom-Technomathematiker (Dipl.-Math. techn.)

Klaus Weber

geboren am 20. Oktober 1964 in Heilbronn am Neckar

Gutachter: Prof. Dr. Ludwig Cromme

Gutachter: Prof. Dr. Volker Nollau

Gutachter: Prof. Dr. Werner Römisch

Gutachter: Dr. Harald Seyfarth

Tag der mündlichen Prüfung: 22. November 2005

Vorwort

Für einen wissenschaftlichen Mitarbeiter einer Universität ist es nicht nur üblich, zu promovieren, sondern es wird von seiner Umgebung geradezu erwartet. So wird es niemanden wundern, dass Ideen zu einer Dissertation während meiner Tätigkeit am Lehrstuhl für numerische und angewandte Mathematik an der Brandenburgischen Technischen Universität (BTU) Cottbus in den Jahren 1994 bis 1999 entstanden sind. Allein, Ideen gibt es mancherlei und sie machen noch keine Doktorarbeit aus. Und so brachte erst die glückliche Projektzusammenarbeit mit der Firma Format in Frankfurt an der Oder im Jahr 1999 die Idee zutage, die genug Substanz und Anspruch besaß. Nur leider sind Spätzünder im Hochschulrahmengesetz nicht vorgesehen. Mein Arbeitsverhältnis war 1999 beendet und im Januar 2000 nahm ich eine Tätigkeit in der Entwicklung von Systemen zur Ertragssteuerung bei Lufthansa Systems in Berlin auf. Der guten Idee aus Cottbus tat das keinen Abbruch und die Arbeit entstand nach Feierabend, am Wochenende und im Urlaub. Ich verdanke es der Eselsgeduld meiner Ehefrau Gabriela und meines ehemaligen Cottbuser Chefs, Prof. Dr. Cromme sowie des aufmunternden Zuspruchs von Freunden, Bekannten, Kollegen und meiner Eltern, dass diese Ochsentour nicht im Graben endete. Am Ende ist die Freude groß und ebenso der Wunsch, all jenen Beteiligten zu danken.

Besonderer Dank gebührt meinem geschätzten chinesischen Freund und ehemaligen Cottbuser Kollegen Sun Zhao hao. Sein persönliches Vorbild lehrte mich: „Mit Geduld und Selbstvertrauen sind alle Schwierigkeiten zu meistern.“ Mit seiner Ankunft am Lehrstuhl begann die fruchtbarste und schöpferischste Zeit meiner Arbeit an der BTU Cottbus. Dankbar bin ich auch den Kolleginnen und Kollegen der BTU, die durch ihre alltägliche Unterstützung, Kollegialität und Freundlichkeit, ein produktives und angenehmes Arbeiten und Zusammenarbeiten ermöglichten. Namentlich genannt seien Silke Büttner, Stephan Würll und Frau Schaarschmidt sowie die Mitarbeiterinnen und Mitarbeiter des Sprachenzentrums, der Universitätsbibliothek und insbesondere der Fernleihstelle. Förderlich war auch die Erlaubnis meines ehemaligen Arbeitgebers Lufthansa Systems Berlin, zahlreiche angesammelte Überstunden in einem längeren Urlaub abzubauen, der meine Arbeit weiter voran brachte. Auch dafür bin ich dankbar.

Meinem Doktorvater, Herrn Prof. Dr. Cromme bin ich sehr dankbar für das in mich gesetzte Vertrauen, das durch die lange Zeit der Anfertigung dieser Arbeit auf eine große Geduldsprobe gestellt worden ist; auch für die zahlreichen Diskussionen und die vielen wertvollen Hinweise und Anmerkungen. Ebenso hilfreich waren die ausführlichen Gutachten und kritischen Hinweise von Herrn Prof. Dr. Nollau, Technische Universität Dresden und Herrn Prof. Dr. Römisch, Humboldt-Universität zu Berlin sowie von Herrn Dr. Seyfarth von der Firma ELAXY Format in Frankfurt (Oder). Auch für Ihre Bereitschaft, die umfangreiche Arbeit zu begutachten, sage ich Ihnen vielmals Dank! Ebenfalls dankbar bin ich Frau Dr. Gröwe-Kuska, früher Mitarbeiterin von Herrn Prof. Dr. Römisch, und meinen ehemaligen Kollegen Dr. Jiri Sustal und Dr. Frieder Kaden von der BTU für ihre kritischen Anmerkungen und Anregungen. Und schließlich danke ich auch den Herren Göhringer und Gäbler, die mich als ABM-Kräfte am Lehrstuhl tatkräftig unterstützt haben.

Es ist schwierig, Worte für Jene zu finden, die in ferner Vergangenheit den Keim gelegt haben und die mich mit Liebe und Vertrauen begleitet haben. Meinen Eltern und meiner lieben Ehefrau Gabriela sei diese Arbeit zum Dank gewidmet.

Rostock, im Dezember 2005

Klaus Weber

Inhaltsverzeichnis

	Abbildungsverzeichnis	IX
	Tabellenverzeichnis	XIII
	Verzeichnis der Definitionen	XV
	Verzeichnis der Sätze	XIX
	Verzeichnis der Algorithmen	XXI
1	Einleitung	1
1.1	Unschärfe und unscharfe Verfahren	3
1.2	Datenbank-Marketing und AkquiSys	5
1.3	Aufbau der Arbeit, Konventionen	10
2	Unscharfe Logik	15
2.1	Was ist unscharfe Logik?	15
2.2	Zur Kritik an der unscharfen Logik	31
2.3	Unscharfe Logik und mehrwertige Logik	33
2.4	Unscharfe Logik und Wahrscheinlichkeitsrechnung	35
2.5	„Soft computing“ und „Computing with words“	39
3	Modellbasierte Kundengewinnung	41
3.1	Kundengewinnung mit Akquiseplänen	42
3.1.1	Akquisepläne des Software-Werkzeugs AkquiSys	43
3.1.2	Formale Beschreibung von Akquiseplänen	44
3.1.3	Stärken und Schwächen von Akquiseplänen	48
3.2	Mathematische Modellierung von Akquiseplänen	49
3.2.1	Wahrscheinlichkeitsraum	52
3.2.2	Wahrscheinlichkeiten und ihre Zusammenhänge	54
3.2.2.1	Präsenzwahrscheinlichkeiten	54
3.2.2.2	Übergangswahrscheinlichkeiten	57
3.2.2.3	Zusammenhang zwischen den Wahrscheinlichkeiten	62
3.2.2.4	Dauer der Akquise	68
3.2.3	Ökonomische Größen	72
3.2.3.1	Ökonomische Größen für einzelne Knoten	75
3.2.3.2	Ökonomische Größen für den ganzen Akquiseplan	76
3.3	Simulation der Kundenakquisition mit Akquiseplänen	77
3.3.1	Bereitzustellende Informationen	77
3.3.2	Implementierung des Verfahrens	78
3.3.3	Durchführung und Auswertung der Simulation – Fallbeispiele	85
3.4	Unscharfe stochastische Optimierung mit Akquiseplänen	88

4	Unscharfe stochastische Optimierung – Optimale Beschickung eines Akquiseplans . .	91
4.1	Behandlung der Unschärfe	91
4.1.1	Grundlagen	91
4.1.2	Erstes Optimierungsproblem	96
4.2	Erwartungswertansatz und Optimierung mit zufälligen Nebenbedingungen . . .	99
4.2.1	Grundlagen	100
4.2.2	Erstes Optimierungsproblem	105
4.2.2.1	Erstes Optimierungsproblem als Erwartungswertproblem	105
4.2.2.2	Optimierung mit zufälligen Nebenbedingungen	107
4.2.3	Zweites Optimierungsproblem	113
4.2.3.1	Zweites Optimierungsproblem als Erwartungswertproblem . . .	117
4.2.3.2	Optimierung mit zufälligen Nebenbedingungen	118
4.2.4	Drittes Optimierungsproblem	124
4.3	Unscharfes stochastisches Mehrgüterfluss-Problem	125
4.3.1	Modellierungsgrundlagen	127
4.3.2	Zufallsvariable	127
4.3.3	Erstes und zweites Optimierungsproblem als ganzzahliges stochastisches lineares Optimierungsproblem	131
4.3.3.1	Bilanzgleichungen der Kundenzahlen	132
4.3.3.2	Erstes Optimierungsproblem	134
4.3.3.3	Zweites Optimierungsproblem	136
4.3.4	Zweites Optimierungsproblem als Mehrgüterfluss-Problem	138
4.3.4.1	Grundlagen	138
4.3.4.2	Maximaler-Fluss-Problem mit unbequemen Nebenbedingungen	141
4.3.4.3	Diskussion der Bündel-Nebenbedingungen	148
4.3.4.4	Szenarienbasiertes deterministisches Äquivalent	155
4.3.4.5	Erweitertes Lagrange-Verfahren	158
4.3.5	Generierung von Zufallszahlen	165
4.4	Fallbeispiel	168
4.4.1	Implementierung des Verfahrens	169
4.4.1.1	Parameterdatei	170
4.4.1.2	Definitionsdateien	175
4.4.1.3	Ausgabe-Dateien	177
4.4.1.4	Konsolenausgabe	182
4.4.1.5	Beschreibung des Computerprogramms	183
4.4.2	Untersuchungsprogramm	186
4.4.2.1	Lösung ohne stochastische Nebenbedingungen	190
4.4.2.2	Lösung mit stochastischen Nebenbedingungen	194
4.4.2.3	Anwendung des Lagrange-Verfahrens	206
5	Mathematische Modellierung dynamischer Akquisepläne	217
5.1	Stärken und Schwächen der Kundengewinnung mit Akquiseplänen	217

5.2	Erweiterte Akquisepläne	218
5.3	Dynamische Akquisepläne	221
5.3.1	Notwendigkeit dynamischer Akquisepläne	222
5.3.2	Definition dynamischer Akquisepläne	223
5.4	Unschärfe stochastische Optimierung mit dynamischen Akquiseplänen	237
6	Mehrstufige stochastische Entscheidungsprozesse in unscharfer Umgebung	239
6.1	Grundbegriffe der dynamischen Optimierung	239
6.2	Dynamische Optimierung in unscharfer Umgebung	243
6.3	Methodik der dynamischen Optimierung mehrstufiger Entscheidungsprozesse mit fester Stufenzahl	248
6.3.1	Deterministischer Entscheidungsprozess mit fester Stufenzahl	248
6.3.2	Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl	251
6.3.3	Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl in unscharfer Umgebung	252
6.4	Dynamische Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit	257
6.4.1	Grundlagen	258
6.4.2	Zulässige Politiken	264
6.4.3	Optimale Politik	275
6.4.4	Verallgemeinerung für andere t-Normen	293
6.4.4.1	Erweiterung der Grundlagen	294
6.4.4.2	Verallgemeinerung der Hauptsätze	297
6.5	Praktische Berechnung einer optimalen Politik	299
6.5.1	Optimierung durch vollständige Aufzählung	300
6.5.2	Optimierung durch Iteration	301
6.5.3	Zur Erreichbarkeit von Endzuständen und einigen anderen praktischen Fragen	303
6.5.3.1	Erreichbarkeit von Endzuständen	303
6.5.3.2	Mehrfache Anwendung gleicher Aktionen	307
6.5.3.3	Granularität der Merkmalsvariablen	307
6.6	Fallbeispiele: Dynamische Akquisepläne	308
6.6.1	Aufbau eines einfachen dynamischen Akquiseplans	309
6.6.2	Aufbau eines komplexen dynamischen Akquiseplans	313
6.6.2.1	Erzeugung der stochastischen Matrizen der Übergangswahrscheinlichkeiten	315
6.6.2.2	Erzeugung der unscharfen Nebenbedingungen	317
6.6.2.3	Erzeugung des unscharfen Zieles	321
6.6.3	Implementierung der Verfahren	322
6.6.3.1	Einlesen und Ausgabe der Daten	323
6.6.3.2	Verfahren zur Bestimmung der Zulässigkeit einer Politik	325

6.6.3.3	Verfahren zur Berechnung der unscharfen Entscheidung einer Politik	327
6.6.3.4	Verfahren zur Berechnung einer optimalen Politik	331
6.6.4	Bestimmung der Zulässigkeit einer Politik und ihrer unscharfen Entscheidung	332
6.6.4.1	Einfacher dynamischer Akquiseplan	332
6.6.4.2	Komplexer dynamischer Akquiseplan	334
6.6.5	Lösung des Optimierungsproblems	336
6.6.5.1	Laufzeitverhalten	336
6.6.5.2	Konvergenzverhalten	337
6.6.5.3	Abhängigkeit der Lösung vom Startwert	340
6.6.5.4	Abhängigkeit der Lösung von der verwendeten t-Norm	341
6.6.5.5	Fachliche Interpretation der optimalen Politiken	342
7	Überblick, Bewertung und Ausblick	347
7.1	Rückblick und Bewertung	347
7.2	Ausblick auf mögliche Erweiterungen	357
7.2.1	Simulation der Kundenakquisition mit Akquiseplänen	357
7.2.2	Optimale Beschickung eines Akquiseplans	358
7.2.3	Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung	358
7.2.4	Softwareprodukt-Prototyp	360
7.3	Anwendungsgebiete	361
	Literaturverzeichnis	363

Abbildungsverzeichnis

Abbildung 1.1:	Zeitliche Entwicklung des Aufwands für organisationsbezogene und persönlichkeitsbezogene Teilprozesse bei der Kundenakquise.....	9
Abbildung 2.1:	Beispiel der unscharfen Menge $R \in IF(U)$ zur Definition der unscharfen Variablen „mittel“ bzw. „mittlerer Fahrzeugabstand“.....	29
Abbildung 2.2:	Die unscharfen Mengen $R(\text{klein})$, $R(\text{mittel})$ und $R(\text{groß})$ der linguistischen Werte „klein“, „mittel“ und „groß“ der linguistischen Variablen „Fahrzeugabstand“.....	30
Abbildung 2.3:	Die unscharfen Mengen $R(\text{klein oder mittel})$ und $R(\text{mittel oder groß})$	30
Abbildung 3.2:	Formale Darstellung des Akquiseplans aus Abb. 3.1.....	44
Abbildung 3.1:	Beispiel eines mit <i>AkquiSys</i> erstellten und visualisierten Akquiseplans...	44
Abbildung 3.3:	Ein Weg im Graphen aus Abb. 3.2.....	45
Abbildung 3.4:	Darstellung des Akquiseplans aus Abb. 3.2 gemäß Def. 3.1.....	46
Abbildung 3.5:	Beispiel: Akquiseplan eines Versicherungsagenten.....	50
Abbildung 3.6:	<i>Zeitliche Verteilungsfunktionen des Knotenwechsels</i> $\phi_{ij}(t)$, $t \in \mathbb{N}_0$ für i, j mit $p_{ij} > 0$;	82
Abbildung 3.7:	Zeitliche Entwicklung der <i>Präsenzwahrscheinlichkeiten</i> $p_{N-1}(t)$ (rot) und $p_N(t)$ (blau) sowie ihrer Summe (schwarz) für den Akquiseplan in Abb. 3.5.....	83
Abbildung 3.8:	Zeitliche Entwicklung des <i>Gewinns</i> für den Akquiseplan in Abb. 3.5....	84
Abbildung 3.9:	<i>Maple</i> -Hauptprogramm in <i>simulation.mws</i>	85
Abbildung 3.10:	Vergleich der Fallbeispiele 1 – 4 (siehe Tab. 3.2): Zeitliche Entwicklung des Gewinns $g_{AP}[0, t]$ für den Akquiseplan in Abb. 3.5.....	88
Abbildung 3.11:	Beispiel einer unscharfen Kapazitätsbeschränkung.....	89
Abbildung 4.1:	Beispiel einer unscharfen Kapazitätsschranke.....	95
Abbildung 4.2:	Beispiel für Hilfssatz 4.5: Die Verteilungsfunktionen der Binomialverteilungen $B_{10, 0,3}$, $B_{10, 0,5}$ und $B_{10, 0,7}$	110
Abbildung 4.3:	Beispiel für Hilfssatz 4.6: die Verteilungsfunktionen der Binomialverteilungen $B_{5, 0,7}$, $B_{10, 0,7}$ und $B_{15, 0,7}$	111
Abbildung 4.4:	Akquiseplan AP mit $A_q = A_1$, $A_n = A_4$ und $A_p = A_5$	144
Abbildung 4.5:	Zeitlicher Akquiseplan AP^T zum Akquiseplan AP in Abb. 4.4 mit Kundenzuführung für $\tau = 0, 1$	145
Abbildung 4.6:	Beispiel eines Akquiseplans zur Erläuterung der Definitionsdateien....	174
Abbildung 4.7:	Definitionsdatei <i>acquirePlan.adj</i> für den Akquiseplan in Abb. 4.6. .	175
Abbildung 4.8:	Definitionsdatei <i>acquirePlan.trs</i> für den Akquiseplan in Abb. 4.6. .	175
Abbildung 4.10:	Definitionsdatei <i>acquirePlan.fcp</i> für den Akquiseplan in Abb. 4.6. .	176
Abbildung 4.9:	Definitionsdatei <i>acquirePlan.tme</i> für den Akquiseplan in Abb. 4.6. .	176
Abbildung 4.11:	Datei mit dem zeitlichen Akquiseplan, <i>acquirePlan.tap</i> für den Akquiseplan in Abb. 4.6.....	179
Abbildung 4.12:	Datei mit dem zeitlichen Akquiseplan <i>acquirePlan.tap</i> für den Akquiseplan in Abb. 4.6 und die Parametereinstellung <code>WRITE_TIMENODE_PREDS_SUCCS = "ON"</code>	180

Abbildung 4.13:	Beispiel einer Ergebnisdatei *.sol (Auszug) für den Akquiseplan in Abb. 4.6 und <code>OPTIMIZATION_MODE = "SCENARIO_LAGRANGE"</code> ;	181
Abbildung 4.14:	Ergebnisdatei <code>aquisePlan.sol</code> für den Akquiseplan in Abb. 3.5, 20 Szenarien und die Zuführungszeitpunkte $\tau = 0, 1, \dots, 5$	182
Abbildung 4.15:	Hauptprogramm <code>main.cpp</code>	187
Abbildung 4.16:	Konsolenausgabe des Programms <code>secondOptimizationProblem.exe</code> für Fall 1	192
Abbildung 4.17:	Fall 10: Untersuchte Kombinationen aus Szenarienanzahl (Parameter <code>SCENARIO_NUMBER</code>) und Dauer der Zuführung von Kunden (Parameter <code>AQUISE_INPUT_DURATION</code>).....	197
Abbildung 4.18:	Grafische Darstellung der Gesamtlaufzeit des Programms und der Laufzeit der <i>CPLEX</i> -Optimierung in Abhängigkeit von der Anzahl der Szenarien für <code>AVGRa_ – AVGRd_</code>	201
Abbildung 4.19:	Grafische Darstellung der Gesamtlaufzeit des Programms und der Laufzeit der <i>CPLEX</i> -Optimierung in Abhängigkeit von der Dauer der Kundenzuführung für <code>AVGRb_</code> , <code>AVGRc_ – AVGRg_</code>	202
Abbildung 4.20:	Grafische Darstellung der Anzahl der zugeführten Kunden (also des Kundendurchsatzes unter Beachtung der Kapazitäts-Nebenbedingungen) in Abhängigkeit von der Dauer der Kundenzuführung bei einer konstanten Anzahl von 20 Szenarien (<code>AVGRb_</code> , <code>AVGRc_ – AVGRg_</code>).....	203
Abbildung 4.21:	Auszug aus <code>AVGTx_analysis.xls</code>	206
Abbildung 4.22:	Auszug aus <code>LGRTx_analysis.xls</code>	215
Abbildung 5.2:	Erweiterter Akquiseplan mit einer in der Akquise integrierten Kundensegmentierung.	218
Abbildung 5.1:	Der Akquise vorgeschaltete Kundensegmentierung.....	218
Abbildung 5.3:	Beispiel: Ausschnitt aus einem <i>erweiterten Akquiseplan</i>	221
Abbildung 5.4:	Beispiel einer Zustandsmenge.	225
Abbildung 5.5:	Beispiel möglicher Zustandsübergänge entsprechend der Kundenreaktionen in einem <i>dynamischen Akquiseplan</i>	228
Abbildung 5.6:	Unschärfe Nebenbedingungen bei der Neukundengewinnung für Lebensversicherungen.	233
Abbildung 5.7:	Dynamischer Akquiseplan mit in die Akquise integrierter Kundensegmentierung und darauf aufbauender Optimierung.	237
Abbildung 6.1:	Unschärfe Entscheidung = „Zusammenkommen“ von unscharfem Ziel und unscharfer Nebenbedingung mit der Minimumbildung als Aggregationsoperator.	245
Abbildung 6.2:	Projektion des Zustandsraumes in die Ebene, die von den Merkmalen „Neuigkeit“ und „Häufigkeit“ aufgespannt wird.	306
Abbildung 6.3:	Auszug aus <code>Beispiel_1.xls</code> , Tabelle P1: Merkmalswerte	309
Abbildung 6.4:	Auszug aus <code>Beispiel_1.xls</code> , Tabelle P3: Übergangswahrscheinlichkeiten	311
Abbildung 6.5:	Auszug aus <code>Beispiel_1.xls</code> , Tabelle C: unscharfe Nebenbedingungen.....	311
Abbildung 6.6:	Auszug aus <code>Beispiel_1.xls</code> , Tabelle G: Unscharfes Ziel;	312
Abbildung 6.7:	Auszug aus <code>Beispiel_1.xls</code> , Tabellen P1 (links) und <code>policy_check</code>	

	(rechts).....	313
Abbildung 6.8:	Auszug aus <code>Beispiel_2.xls</code> , Tabelle P1: Merkmalswerte	314
Abbildung 6.9:	Auszug aus <code>rules_TransProbs.txt</code> : Zu sehen sind die Regeln für das Merkmal <i>Alter</i> in der <i>CLIPS</i> -Sprache.....	318
Abbildung 6.10:	Schematische Darstellung der Datenflüsse (durchgezogene Pfeile) und der Abhängigkeiten im <i>CLIPS</i> -Programm (gestrichelte Pfeile).	319
Abbildung 6.11:	Auszug aus <code>rules_Constraints.txt</code> : Zu sehen sind die Regeln für das Merkmal <i>Alter</i> in der <i>CLIPS</i> -Sprache.....	321
Abbildung 6.12:	Auszug aus <code>rules_Goal.txt</code> : Zu sehen sind die Regeln für die Merkmale <i>Produktinteresse</i> und <i>Alter</i> in der <i>CLIPS</i> -Sprache.	322
Abbildung 6.13:	Schematische Darstellung der Datenflüsse (durchgezogene Pfeile) und der Abhängigkeiten im <i>CLIPS</i> -Programm (gestrichelte Pfeile).	323
Abbildung 6.14:	Tabelle <code>Parameter</code> in den <i>Excel</i> -Dateien <code>Beispiel_1.xls</code> mit Verfahrensparametern.	324
Abbildung 6.15:	Auszug aus dem <i>MATLAB</i> -Programmcode der Funktion <code>checkGivenPolicy</code> in der Datei <code>checkGivenPolicy.m</code>	326
Abbildung 6.16:	Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionsaufrufe (blaue Pfeile) im <i>MATLAB</i> -Programm (grüner Kasten) bei der Überprüfung einer gegebenen oder mehrerer zufälliger Politiken auf Zulässigkeit für den komplexen dynamischen Akquiseplan.....	328
Abbildung 6.17:	Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionenaufrufe (blaue Pfeile) im <i>MATLAB</i> -Programm (grüner Kasten) bei der Berechnung der unscharfen Entscheidung zu einer gegebenen Politik für den komplexen dynamischen Akquiseplan.....	329
Abbildung 6.18:	Auszug aus dem <i>MATLAB</i> -Programmcode der Funktion <code>trafoIter</code> in der Datei <code>trafoIter.m</code>	330
Abbildung 6.19:	Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionenaufrufe (blaue Pfeile) im <i>MATLAB</i> -Programm (grüner Kasten) bei der Berechnung einer optimalen Politik für den komplexen dynamischen Akquiseplan.....	331
Abbildung 6.20:	Auszug aus dem <i>MATLAB</i> -Programmcode der Funktion <code>maxTrafoIter</code> in der Datei <code>maxTrafoIter.m</code>	332
Abbildung 6.21:	<code>Beispiel_1.xls</code> , Tabelle <code>policy_check</code>	333
Abbildung 6.22:	<code>Beispiel_2.xls</code> , Auszug aus der Tabelle <code>policy_check</code> (für die ersten und letzten zehn Zustände).....	335
Abbildung 6.23:	Laufzeitverhalten der <i>MATLAB</i> -Funktion <code>maxTrafoIter</code> beim Ausführen des Programms <code>optimalPolicy</code> auf dem komplexen dynamischen Akquiseplan	337
Abbildung 6.24:	Darstellung des Verlaufs der Größen <code>normDiff</code> (6.112) und <code>normQuot</code> (6.113) für alle hier betrachteten <i>t</i> -Normen und den Anfangsvektor $\omega^{(0)} = 1 \in \mathbb{R}^k$	338
Abbildung 6.25:	Darstellung des Verlaufs der Größen <code>normDiff</code> (6.112) und <code>normQuot</code> (6.113) für alle hier betrachteten <i>t</i> -Normen und den Anfangsvektor $\omega^{(0)} = 0 \in \mathbb{R}^k$	339
Abbildung 6.26:	Auszug aus <code>Beispiel_1.xls</code> , Tabelle <code>policy_result</code>	341

Abbildung 6.27: Übersicht der unscharfen Entscheidungen $\omega^{(n)}$ (oben) und der zugehörigen maximierenden Politik $p^{(n)}$ (unten) für die Iterationen der Transformation (6.114) und die t-Normen <i>algebraisches Produkt</i> und <i>Hamacher-Produkt</i>	343
Abbildung 6.28: Auszug aus <i>Beispiel_1.xls</i> , Tabelle <i>result</i> . Optimale Politiken und ihre unscharfen Entscheidungen (gegeben durch die Wertepaare (σ, α)) für alle t-Normen, den Startvektor $\omega^{(n)} = 0$ und $\text{eps} = 1\text{E-}12$ (<i>stop criterion</i>).	344
Abbildung 6.29: Funktionsgraphen der t-Normen <i>algebraisches Produkt</i> , <i>Hamacher-Produkt</i> und <i>Einstein-Produkt</i>	344

Tabellenverzeichnis

Tabelle 2.1:	Beispiel für den Unterschied zwischen Möglichkeit und Wahrscheinlichkeit.	38
Tabelle 3.1:	Übersicht von ökonomischen Größen, die in einer Simulation des Akquise-Prozesses berechnet werden könnten.	73
Tabelle 3.2:	Werte der Ertragsfunktion (Kosten/Erträge) für den Akquiseplan in Abb. 3.5 für vier Fallbeispiele.	85
Tabelle 4.1:	Werte der Umkehrabbildung für die Zugehörigkeitsfunktion in Abb. 4.1.	95
Tabelle 4.2:	Untersuchungsprogramm für die Verfahren zur Lösung des zweiten Optimierungsproblems.	188
Tabelle 4.3:	Quotienten aus Größen des zweiten Optimierungsproblems bzw. des zugehörigen zeitlichen Akquiseplans und der <i>CPLEX</i> -Laufzeit;	202
Tabelle 4.4:	Vergleich der Anzahlen zugeführter Kunden für $t = 0, 1, \dots, 5$ für <i>AVG2_</i> , <i>LGR2d_</i> und <i>LGR2g_</i> .	212
Tabelle 5.1:	Einige Zustände in einem Zustandsraum.	225
Tabelle 5.2:	Einige Zustände in einem Zustandsraum.	233
Tabelle 5.3:	Entscheidungen $\alpha_i \in U$, $i = 1, \dots, 6$ aus Abb. 5.6 und ihre Kosten.	233
Tabelle 5.4:	Beispiel für die Interpretation der Merkmale <i>Neuigkeit</i> und <i>Häufigkeit</i> .	236

Verzeichnis der Definitionen

Definition 1.1:	<i>Unschärfe, unscharfe Verfahren.</i>	5
Definition 1.2:	<i>Datenbank-Marketing.</i>	7
Definition 1.3:	<i>Wissensentdeckung in Datenbanken (kurz: WED).</i>	8
Definition 2.1 („Naive Definition“):	<i>Unscharfe Menge, Zugehörigkeitswert, Grundmenge.</i>	19
Definition 2.2:	<i>Unscharfe Menge (1. Darstellungsform), Grundmenge, Zugehörigkeitswert.</i>	19
Definition 2.3:	<i>Unscharfe Menge (2. Darstellungsform), Grundmenge, Zugehörigkeitsfunktion, Zugehörigkeitswert, Menge der unscharfen Mengen.</i>	20
Definition 2.4:	<i>Träger.</i>	20
Definition 2.5:	<i>α-Schnitt.</i>	21
Definition 2.6:	<i>Normale unscharfe Menge.</i>	21
Definition 2.7:	<i>t-Norm, t-Konorm, s-Norm.</i>	21
Definition 2.8:	<i>Negationsfunktion, strenge Negationsfunktion, involutorische Negationsfunktion.</i>	22
Definition 2.9:	<i>Φ-Operator.</i>	23
Definition 2.10:	<i>Unscharfe Konjunktion, unscharfe Disjunktion, unscharfe Implikation, unscharfe Negation.</i>	24
Definition 2.11:	<i>Unscharfer Durchschnitt, unscharfe Vereinigung, unscharfes Komplement, unscharfes kartesisches Produkt.</i>	26
Definition 2.12:	<i>Erweiterungsprinzip.</i>	27
Definition 2.13:	<i>Unscharfe Variable, unscharfe Beschränkung, Verträglichkeitsgrad.</i>	28
Definition 2.14:	<i>Linguistische Variable, Term-Menge, linguistischer Wert, Basisvariable, syntaktische Regel, semantische Regel.</i>	29
Definition 2.15:	<i>Linguistischer Modifikator.</i>	31
Definition 3.1:	<i>Akquiseplan.</i>	45
Definition 3.2:	<i>Adjazenzmatrix eines Akquiseplans.</i>	47
Definition 3.3:	<i>Mögliche Knotenfolge.</i>	52
Definition 3.4:	<i>Merkmalsraum, Ereignisfeld, Ereignis, Wahrscheinlichkeitsverteilung, Wahrscheinlichkeitsraum.</i>	52
Definition 3.5:	<i>Ereignis „ein Kunde befindet sich zum Zeitpunkt t im Knoten i“.</i>	53
Definition 3.6:	<i>Präsenzwahrscheinlichkeit, Vektor der Präsenzwahrscheinlichkeiten.</i>	54
Definition 3.7:	<i>Bedingte Wahrscheinlichkeit des Knoteneintritts.</i>	55
Definition 3.8:	<i>Wahrscheinlichkeit des Wechsels nach Ankunft im Knoten.</i>	58
Definition 3.9:	<i>Zeitliche Punktwahrscheinlichkeit des Knotenwechsels, zeitliche Verteilungsfunktion des Knotenwechsels.</i>	58
Definition 3.10:	<i>Wahrscheinlichkeit des Wechsels nach Wartezeit im Knoten.</i>	61
Definition 3.11:	<i>Dauer einer Akquise.</i>	68
Definition 3.12:	<i>Endlicher Träger einer zeitlichen Punktwahrscheinlichkeit.</i>	68
Definition 3.13:	<i>(vollständige) Menge der Vorgängerknoten/-aktionen, (vollständige) Menge der Nachfolgerknoten/-aktionen eines Knotens/einer Aktion.</i>	68
Definition 3.14:	<i>Ertragsfunktion.</i>	75
Definition 4.1:	<i>Kundenanzahl zur Zeit t im Knoten A_i für c_0 Anfangskunden, Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden.</i>	92
Definition 4.2:	<i>Unscharfe Kapazitätsschranke.</i>	93
Definition 4.3:	<i>Verallgemeinerte Umkehrabbildung einer monoton abnehmenden Zugehörigkeitsfunktion über IN_0.</i>	94

Definition 4.4:	<i>Scharfes Äquivalent eines linearen Optimierungsproblems mit unscharfer rechter Seite zum Wahrheitsniveau δ.</i>	97
Definition 4.5:	<i>Erstes Optimierungsproblem zum Wahrheitsniveau δ für deterministisches Kundenverhalten.</i>	97
Definition 4.6:	<i>Präsenzwahrscheinlichkeit neuer Kunden, Vektor der Präsenzwahrscheinlichkeiten neuer Kunden.</i>	100
Definition 4.7:	<i>Zufallsgröße der Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden.</i>	102
Definition 4.8:	<i>Erstes Optimierungsproblem als Erwartungswertproblem.</i>	105
Definition 4.9:	<i>Erstes Optimierungsproblem als Optimierungsproblem mit zufälligen Nebenbedingungen.</i>	108
Definition 4.10:	<i>Verallgemeinerte inverse Verteilungsfunktion [69].</i>	109
Definition 4.11:	<i>Erweiterung von Def. 4.6.</i>	114
Definition 4.12:	<i>Zuführungszeitraum, Zufallsgröße der Anzahl neuer Kunden zum Zeitpunkt t im Knoten A_i.</i>	115
Definition 4.13:	<i>Zweites Optimierungsproblem als Erwartungswertproblem.</i>	117
Definition 4.14:	<i>Zweites Optimierungsproblem als Optimierungsproblem mit zufälligen Nebenbedingungen.</i>	119
Definition 4.15:	<i>Drittes Optimierungsproblem als Erwartungswertproblem / als Optimierungsproblem mit zufälligen Nebenbedingungen.</i>	124
Definition 4.16:	<i>Minimaler Abstand eines Knotens von der Quelle, minimaler Abstand eines Knotens von den Senken.</i>	127
Definition 4.17:	<i>Normierte Multinomialverteilung.</i>	129
Definition 4.18:	<i>Kundenverteilung, stochastischer Prozess der Kundenverteilung, Szenario der Kundenverteilung, Stichprobennahme der Kundenverteilung, Stichprobe der Kundenverteilung.</i>	130
Definition 4.19:	<i>Anzahl der Kunden/Kundenanzahl zur Zeit t im Knoten A_i, Anzahl neuer Kunden zur Zeit t im Knoten A_i, Anzahl gegangener Kunden zur Zeit t im Knoten A_i, Anzahl der Kunden, die zur Zeit t vom Knoten A_i zum Knoten A_j gewechselt haben, Anzahl der Kunden, die zur Zeit t im Knoten A_i geblieben sind.</i>	131
Definition 4.20:	<i>Erstes Optimierungsproblem.</i>	135
Definition 4.21:	<i>Zweites Optimierungsproblem.</i>	137
Definition 4.22:	<i>Flussvektor eines Graphen, Fluss einer Kante, Divergenz eines Graphen, Divergenz eines Knotens, Quellenstärke, Senkenstärke, Quelle, Senke, Fluss des Graphen.</i>	138
Definition 4.23:	<i>Minimaler-Kosten-Fluss-Problem.</i>	139
Definition 4.24:	<i>Maximaler-Fluss-Problem.</i>	140
Definition 4.25:	<i>τ-Menge zur Zeit t und zum Knoten A_i.</i>	141
Definition 4.26:	<i>Zeitlicher Akquiseplan.</i>	141
Definition 4.27:	<i>(vollständige) Menge der Vorgängerknoten, (vollständige) Menge der Nachfolgerknoten im zeitlichen Akquiseplan.</i>	144
Definition 4.28:	<i>Knoten-Kanten-Inzidenzmatrix eines (zeitlichen) Akquiseplans.</i>	148
Definition 4.29:	<i>Total unimodular.</i>	149
Definition 5.1:	<i>Merkmalsvariable, Merkmalswert, Merkmalsmenge, Kundendatensatz-Variable, Kundendatensatz, Kundendatensatz-Menge, Standardwert, negKE, posKE.</i>	223
Definition 5.2:	<i>Akquise-Zustandsmenge, Akquise-Zustand (zum Zeitpunkt t), Akquise-Zustandsvariable.</i>	224

Definition 5.3:	<i>Akquise-Entscheidungsmenge, Akquise-Entscheidungsvariable.</i>	226
Definition 5.4:	<i>Akquise-Zustandsübergang, Akquise-Übergangswahrscheinlichkeit.</i>	226
Definition 5.5:	<i>Menge der positiven Akquise-Endzustände, Menge der negativen Akquise-Endzustände, Menge der Akquise-Endzustände, Menge der nicht-beendeten Akquise-Zustände.</i>	229
Definition 5.6:	<i>Akquise-Ertragsfunktion, unscharfes Akquise-Ziel, unscharfes Akquise-Ziel zur Ertragsfunktion.</i>	230
Definition 5.7:	<i>Unscharfe Akquise-Nebenbedingung.</i>	232
Definition 5.8:	<i>Stationäre Akquise-Entscheidungsregel, stationäre Politik.</i>	234
Definition 5.9:	<i>Dynamischer Akquiseplan.</i>	235
Definition 6.1:	<i>Dynamisches System, Zustandsvektor, Dimension.</i>	239
Definition 6.2:	<i>Mehrstufiger deterministischer/stochastischer Entscheidungsprozess, Zustand, Anfangszustand, Endzustand, Stufenzahl, Zustandsvariable, Zustandsmenge/-raum, Zustandsübergang, Entscheidungsvariable, zulässige Entscheidung, Entscheidungsmenge/-raum, Entscheidung, Zielfunktion, Übergangswahrscheinlichkeit.</i>	240
Definition 6.3:	<i>Politik, stationäre Politik, optimale Politik, Menge der Politiken, Politikraum (für scharfe Entscheidungsprozesse).</i>	241
Definition 6.4:	<i>Unscharfes Ziel.</i>	243
Definition 6.5:	<i>Unscharfe Nebenbedingung.</i>	243
Definition 6.6:	<i>Unscharfe Entscheidung, maximierende Entscheidung.</i>	244
Definition 6.7:	<i>Unscharfe Entscheidung (Verallgemeinerung von Def. 6.6).</i>	245
Definition 6.8:	<i>Politik, stationäre Politik, optimale Politik (für unscharfe Entscheidungsprozesse).</i>	246
Definition 6.9:	<i>Mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit.</i>	253
Definition 6.10:	<i>Mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, Menge der Endzustände, Menge der nicht-beendeten Zustände.</i>	258
Definition 6.11:	<i>Politikvektor, Vektor der unscharfen Nebenbedingungen, unscharfer Entscheidungsvektor, unscharfer Zielvektor.</i>	262
Definition 6.12:	<i>Erreichbarkeit unter der Politik π, Zulässige Politik, Menge der zulässigen Politiken.</i>	265
Definition 6.13:	<i>Besser-Relation der Politiken, optimale Politik.</i>	275
Definition 6.14:	<i>Ähnlichkeit von Politiken.</i>	277
Definition 6.15:	<i>Politik-Äquivalenzklasse, Politik-Faktormenge, Besser-Relation der Politik-Äquivalenzklassen.</i>	277
Definition 6.16:	<i>Entscheidungsvektor einer Politik-Äquivalenzklasse.</i>	279
Definition 6.17:	<i>Transformationen T_{π}^t und T^t.</i>	297

Verzeichnis der Sätze

Satz 2.1:	22
Satz 2.2:	23
Satz 2.3:	26
Satz 2.4:	27
Satz 3.1:	47
Hilfssatz 3.2:	55
Hilfssatz 3.3:	57
Hilfssatz 3.4:	59
Satz 3.5:	59
Satz 3.6:	60
Hilfssatz 3.7:	62
Satz 3.8:	62
Satz 3.9:	64
Satz 3.10:	65
Satz 3.11:	66
Satz 3.12:	67
Hilfssatz 3.13:	69
Hilfssatz 3.14:	69
Hilfssatz 3.15:	70
Satz 3.16:	71
Satz 4.1:	98
Satz 4.2:	101
Satz 4.3:	104
Satz 4.4:	106
Hilfssatz 4.5:	109
Hilfssatz 4.6:	110
Satz 4.7:	111
Satz 4.8:	121
Hilfssatz 4.9:	132
Satz 4.10:	149
Satz 4.11:	149
Satz 4.12:	149
Satz 4.13:	151
Satz 4.14:	158
Hilfssatz 4.15:	166
Hilfssatz 4.16:	167
Satz 6.1:	242
Satz 6.2:	249
Satz 6.3:	252
Satz 6.4:	256
Hilfssatz 6.5:	265
Satz 6.6:	267
Korollar 6.6.1:	268
Hilfssatz 6.7:	271
Satz 6.8 (Weissingerscher Fixpunktsatz):	271
Satz 6.9:	272
Hilfssatz 6.10:	276

Hilfssatz 6.11:	276
Hilfssatz 6.12:	277
Satz 6.13:	278
Satz 6.14:	279
Korollar 6.14.1:	279
Hilfssatz 6.15:	280
Satz 6.16:	282
Satz 6.17:	284
Hilfssatz 6.18:	286
Hilfssatz 6.19:	286
Satz 6.20:	286
Korollar 6.20.1:	292
Satz 6.21:	295
Satz 6.22:	297
Hilfssatz 6.23:	298
Satz 6.24:	298
Satz 6.25:	298
Satz 6.26:	299

Verzeichnis der Algorithmen

Algorithmus 3.1: Algorithmus zur iterativen Berechnung der Präsenzwahrscheinlichkeiten $p(t)$, $t = 0, \dots, t_{\max} - 1$ gemäß Satz 3.12.....	79
Algorithmus 4.1: Lösung des Optimierungsproblems gemäß Def. 4.14 durch vollständige Aufzählung.	122
Algorithmus 4.2: Unterprogramm <i>BerechneDimC(r)</i>	123
Algorithmus 4.3: Unterprogramm <i>IstZulässig(c^{tin})</i>	123
Algorithmus 4.4: Unterprogramm <i>Summe(c^{tin})</i>	123
Algorithmus 4.5: Lösung des dritten Optimierungsproblems.....	126
Algorithmus 4.6: Algorithmus zur Lösung von (4.131) mit dem erweiterten Lagrange-Verfahren.....	160
Algorithmus 4.7: Algorithmus zur Erzeugung eines multinomialverteilten Zufallsvektors. .	168
Algorithmus 6.2: „Naive“ vollständige Aufzählung aller Politiken.	300
Algorithmus 6.1: Unterprogramm <i>Initialisierung</i>	300
Algorithmus 6.3: Optimierung durch vollständige Aufzählung aller Politiken.	301
Algorithmus 6.4: Optimierung durch Iteration.	302
Algorithmus 6.5: Unterprogramm <i>Epsilonkonstante</i>	303
Algorithmus 6.6: Optimierung durch Iteration (Alg. 6.4) mit verbesserter Abbruchbedingung.	304

1 Einleitung

Der Inhalt dieser Arbeit ist anwendungsorientiert. Sie entwickelt theoretische und algorithmische Grundlagen für ein System zur Entscheidungsunterstützung bei der Kundenakquisition im Marketing. Besondere Beachtung finden dabei

- die zeitliche *Simulation* des Kundenverhaltens sowie ökonomischer Kenngrößen,
- die *Optimierung der Anzahl der Kunden*, die mit einer Marketingstrategie bearbeitet werden,
- die *Optimierung der Akquiseaktionen* einer Marketingstrategie.

Neben *Simulation* und *Optimierung* steht die *Modellbildung* im Mittelpunkt. Sie berücksichtigt zum einen die *Stochastik* des Kundenverhaltens und zum anderen die *Unschärfe* verschiedener Elemente der Kundenakquisition: der Kapazitätsbeschränkung für Akquiseaktionen, des Zieles der Kundenakquisition sowie der Bewertung, welche Aktionen für welche Kunden gut geeignet sind. Bei der Modellierung wird schrittweise vorgegangen:

- Formalisierung von Marketingstrategien mit Hilfe von *Akquiseplänen* und einem *Wahrscheinlichkeitsraum*, mit dem praktisch handhabbare Verfahren zur *Simulation* des Kundenverhaltens hergeleitet werden können;
- Formulierung des *unscharfen stochastischen Optimierungsproblems* der optimalen Beschi-ckung von Akquiseplänen;
- Erweiterung von Akquiseplänen zu *dynamischen Akquiseplänen*, mit denen eine Marketingstrategie als eine Akquise-Entscheidungsregel eines *mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit* dargestellt und optimiert werden kann.

Der Schwerpunkt der Arbeit liegt auf der Formulierung der Optimierungsprobleme und der Untersuchung ihrer Lösbarkeit. Als erste große thematische Einheit zur unscharfen Optimierung wird die Lösung des Problems der *optimalen Beschickung von Akquiseplänen* behandelt. Dabei werden Ansätze der unscharfen linearen Optimierung, über die z. B. [99, 138] eine umfassenden Übersicht geben, erweitert. Die Modellierung der Stochastik des Kundenverhaltens führt schließlich zur Erzeugung des deterministischen Äquivalents mit Hilfe von *Szenarien*. In gleicher Weise wird das *scharfe Äquivalent* des unscharfen Problems gewonnen. Die Strukturuntersuchung des Optimierungsproblems führt zu einer Erweiterung des Akquiseplan-Modells zum *zeitlichen Akquiseplan* und zur Beschreibung des Problems als ein *Mehrgüterfluss-Problem*. Als Lösungsverfahren werden

- die *Mittelwertbildung* über die Lösungen von linearen Optimierungsproblemen für unterschiedliche Szenarien sowie
- ein erweitertes *Lagrange-Verfahren*, das alle Szenarien zugleich berücksichtigt,

entwickelt, beschrieben und untersucht.

Die wesentlichen theoretischen Ergebnisse der Arbeit werden in der zweiten großen thematischen Einheit zur unscharfen Optimierung gewonnen, die die *Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Pro-*

zess-Endzeit behandelt. Sie basiert auf der Arbeit von Bellman und Zadeh: *Decision-Making in a Fuzzy Environment* [11] und deren Fortführung, wie sie allseitig in [86] beschrieben ist. Während die bekannten Ansätze stochastische Entscheidungsprozesse mit vorab festgelegter Stufenanzahl betrachten, ist in der vorliegenden Arbeit die Prozess-Endzeit *implizit* durch Erreichen eines Endzustandes bestimmt. Die theoretische Untersuchung dieses Problemtyps führt schließlich zu einem praktisch einsetzbaren Verfahren. Eine andere Erweiterung der in der Literatur beschriebenen Behandlung dieser Optimierungsprobleme ist die Untersuchung der in der Modellierung benutzten *t-Norm*. Als Ergebnis kann gezeigt werden, dass alle wesentlichen Aussagen zur Lösung des Problemtyps für eine ganze Klasse von t-Normen gültig sind.

Die Geschichte der vorliegenden Arbeit beginnt mit der Zusammenarbeit des *Lehrstuhls für Numerische und Angewandte Mathematik* an der *Brandenburgischen Technischen Universität Cottbus* mit der Firma *Format* in Frankfurt/Oder¹⁾, die Software-Produkte für den Finanzdienstleistungsbereich entwickelt. Trotz der Allgemeinheit im wissenschaftlichen Gehalt ist die Arbeit deshalb von den Besonderheiten des Finanzdienstleistungs-Marketing geprägt. Ein wichtiger Bezugspunkt der Arbeit ist das Software-Werkzeug *AkquiSys* [30, 146, 147], mit dem der Verlauf einer Kundenakquisition geplant, organisiert und ausgewertet werden kann. Aus einer größeren Anzahl von Aufgabenstellungen zur Erweiterung von *AkquiSys* [32, 33] entwickelten sich im Verlauf der Projektzusammenarbeit verschiedene Richtungen heraus, an denen der Verfasser mitgewirkt hat. Ein großer Teil davon ist Gegenstand dieser Arbeit, auf andere wird an entsprechender Stelle nur verwiesen. Nicht näher eingegangen wird auf Entwicklungen im Bereich intelligenter Agenten („intelligent agents“) [51, 52, 53, 166], die ebenfalls auf diese Zusammenarbeit zurückgehen²⁾. Die hier entwickelten Modelle und Verfahren können zum Gebiet des finanzwirtschaftlichen Ingenieurwesens („financial engineering“) gerechnet werden, das aus dem Versuch heraus entstanden ist, Fragestellungen im Finanzwesen mit den Methoden des Ingenieurwesens zu bearbeiten [137]. Aufgrund der Allgemeinheit der gefundenen Verfahren gehen die Anwendungsmöglichkeiten der Ergebnisse dieser Arbeit aber über den Bereich des Finanzwesens hinaus.

Vorarbeiten des Verfassers zu den hier entwickelten Verfahren sind unter anderem in folgenden Arbeiten veröffentlicht: zur Simulation in [34, 162], zur optimalen Beschickung von Akquiseplänen in [172] und zur Berechnung einer optimalen Akquise-Entscheidungsregel für dynamische Akquisepläne in [161, 163, 164].

Alle hier entwickelten Verfahren sind in Algorithmen umgesetzt und in höheren Programmiersprachen programmiert worden:

- die zeitliche Simulation des Kundenverhaltens und ökonomischer Kenngrößen einer Kundenakquise in *Maple*,
- die Optimierung der Beschickung von Akquiseplänen in *C++* unter Verwendung von *CPLEX* und

1. Das Unternehmen heißt jetzt *ELAXY Format GmbH* und ist ein Teil der *ELAXY GmbH*, Coburg; <http://www.elaxy.de>.

2. Einen einführenden Überblick gibt [122].

- die Berechnung einer optimalen Politik von mehrstufigen stochastischen Entscheidungsprozessen in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit in *Matlab*³⁾.

Die praktische Tauglichkeit der Verfahren wird anhand zahlreicher Fallbeispiele illustriert und untersucht. Alle Programme und die Fallbeispiele sind in dieser Arbeit soweit beschrieben, wie es zu ihrem Verständnis nötig ist. Der Quellcode der Programme und alle für die Fallbeispiele nötigen Daten sowie ihre Ergebnisse sind über die Homepage des Verfassers zugänglich⁴⁾. Die Entscheidung für die verwendeten Programmiersprachen ist der Zweckmäßigkeit ihres Einsatzes, der Einfachheit der Programmierung und auch der Neugier des Verfassers geschuldet.

Der folgende Abschnitt 1.1 führt in den Begriff der *Unschärfe* ein. Der Wunsch, sie formal beschreiben zu können, führte zur Entwicklung der *Theorie unscharfer Mengen* durch Zadeh im Jahre 1965 [177], die auch dieser Arbeit zugrunde liegt. Abschnitt 1.2 führt in das *Datenbank-Marketing* ein, erklärt in diesem Zusammenhang das Software-Werkzeug *AkquiSys* und motiviert daraus die Gegenstände dieser Arbeit. Und schließlich gibt Abschnitt 1.3 einen Überblick über den Aufbau der nachfolgenden Kapitel.

1.1 Unschärfe und unscharfe Verfahren

Die unscharfe Logik wurde 1965 von dem in Baku gebürtigen US-amerikanischen Elektrotechnik-Professor Lotfi A. Zadeh mit dem Zeitschriftenartikel *Fuzzy Sets* [177] eingeführt. Auf die Bedeutung dieses Begriffes und seine Beziehung zu anderen wissenschaftlichen Disziplinen bzw. seine Abgrenzung ihnen gegenüber geht Kapitel 2 näher ein. Zuvor ist es notwendig, den Begriff der Unschärfe zu erklären.

In seinem Zeitschriftenartikel *Fuzzy Sets* motiviert Zadeh die Einführung unscharfer Mengen mit der Schwierigkeit, Gegenstände der physischen Welt und des Denkens immer eindeutig einer Klasse, das heißt einem Oberbegriff zuzuordnen [177]. So sei die Zuordnung von Hunden zur Klasse der Tiere eindeutig, bei Seesternen und Bakterien treffe dies jedoch nicht zu. Dieselbe Schwierigkeit tritt beim Verhältnis der Zahl 10 gegenüber der Klasse der reellen Zahlen, die viel größer als 1 sind, auf. Zadeh stellt fest, dass die Schwierigkeit der Zuordnung von der Ungenauigkeit der Definition solcher Begriffe herrührt, dass ihnen jedoch große Bedeutung im menschlichen Denken zukommt, vor allem in den Bereichen der Wahrnehmung, Verständigung, Information und Abstraktion. Diese Beobachtung verallgemeinert er in einer späteren Arbeit [178] zum so genannten *Inkompatibilitätsprinzip*:

„Stated informally, the essence of this principle is that as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.“

Die mit dem Inkompatibilitätsprinzip beschriebene Beobachtung ist nicht neu und findet nicht erst seit Zadeh Beachtung. Mit Bezug auf die Mathematik formulierte z. B. Einstein im Jahre 1921:

3. Anmerkungen zu den Programmiersprachen finden sich in den Abschnitten, wo die Programme beschrieben werden.

4. <http://vieta.math.tu-cottbus.de/~klweber>

„Insofern sich die Sätze der Mathematik auf die Wirklichkeit beziehen, sind sie nicht sicher; und insofern sie sicher sind, beziehen sie sich nicht auf die Wirklichkeit.“ [45]

Die Schwierigkeit bei der Abgrenzung von Gegenständen der realen Welt und des Denkens widerspiegelt sich auch im Bereich der Logik bei der Zuordnung von Wahrheitswerten zu Aussagen:

„Ich kann ohne Widerspruch annehmen, dass meine Anwesenheit in Warschau in einem bestimmten Zeitmoment des nächsten Jahres, z. B. mittags den 21. Dezember, heutzutage weder im positiven noch im negativen Sinne entschieden ist. Es ist somit möglich, aber nicht notwendig, dass ich zur angegebenen Zeit in Warschau anwesend sein werde. Unter dieser Voraussetzung kann die Aussage: „ich werde mittags den 21. Dezember nächsten Jahres in Warschau anwesend sein“, heutzutage weder wahr noch falsch sein.“ [106]

Diese Beobachtung führte zur Herausbildung zunächst der drei- und später der mehrwertigen Logik, auf die im Abschnitt 2.3 noch eingegangen wird.

In der Literatur wird der Begriff der Unschärfe häufig im Zusammenhang mit Begriffen wie *Unsicherheit* [195] und *Vagheit* [29], *Ungewissheit* und *Ungenauigkeit* [6] bzw. den entsprechenden englischen Begriffen „*uncertainty*“, „*vagueness*“ [95] oder „*imprecision*“ [181] erwähnt oder mit diesen Begriffen umschrieben. In [95] wird von „*partial ignorance*“ gesprochen und als deren wichtigsten Formen von „*vagueness*“ und „*uncertainty*“:

„The notion of vagueness refers to the ambiguity (in its colloquial meaning) remaining due to the hazy content of some datum concerning a questionable facet of the actual problem. Vagueness arises whenever a datum, although its meaning is not in doubt, lacks the desired precision. Examples may be the result of some measuring process which yields that some electrical potential is in the range from 10V to 12V ...

Uncertainty, on the other hand, corresponds to a human being's valuation of some datum, reflecting his or her faith or doubt in its source. This concept covers those cases in which the actual state of affairs or process is not completely determined but where we have to rely on some human expert's subjective preferences among the different possibilities. ... Another example is a physician's uncertainty concerning the type of cancer a patient suffers from.“

Zimmermann wiederum verwendet für die hier zitierten Inhalte der Begriffe „*vagueness*“ und „*uncertainty*“ die Bezeichnungen „*intrinsic fuzziness*“ und „*informational fuzziness*“ [194].

Beschreibungen von Unschärfe werden in der Literatur häufig auch gegenüber *Wahrscheinlichkeit* bzw. „*randomness*“, „*probability*“ abgegrenzt. Diese Abgrenzung ist in der wissenschaftlichen Auseinandersetzung um die Theorie unscharfer Mengen von besonderer Bedeutung, weshalb in Abschnitt 2.4 näher darauf eingegangen wird. Definitionen von Unschärfe bedienen sich üblicherweise der Gegenüberstellung und Abgrenzung zu den genannten Begriffen. Die Weitschweifigkeit vieler Definitionen offenbart die Schwierigkeit, die die Beschreibung des Begriffs bereitet. Die folgende nicht-mathematische Definition versucht, die in oben genannten Zitaten beschriebenen Beobachtungen zusammenzufassen.

Definition 1.1: *Unschärfe, unscharfe Verfahren.*

Unschärfe ist eine gedankliche Widerspiegelung von Gegenständen der objektiven Welt und des menschlichen Denkens, die auf Begriffen beruht, die sich nicht gegenseitig ausschließen.

Unscharfe Verfahren operieren mit solchen Begriffen. Unschärfe und unscharfe Verfahren entspringen der praktischen Notwendigkeit, sich ein Bild von der durch stetige Entwicklung und Wechselwirkung gekennzeichneten Welt zu machen. □

Der erste Satz der Definition grenzt Unschärfe gegen jede Art dichotomischer Begriffsbildung ab, die dem Satz vom ausgeschlossenen Widerspruch („Eine Aussage ist nicht zusammen mit ihrem Negat wahr.“) und vom ausgeschlossenen Dritten („Zwischen einer Aussage und ihrem Negat gibt es nichts Drittes.“) gehorchen [105, 144]. Der zweite Satz der Definition betont den pragmatischen Aspekt der Begriffsbildung, das heißt Unschärfe ist keine Kopfgeburt, sondern entspringt der Notwendigkeit, Beobachtungen zu beschreiben, die mit anderen als unscharfen Begriffen nur mit Schwierigkeiten oder gar nicht zu fassen wären⁵⁾.

Unscharfe Verfahren beschränken sich nicht nur auf die Theorie unscharfer Mengen, die in Abschnitt 2.1 eingeführt wird, und ihre jüngeren Geschwister, „soft computing“ und „computing with words“, die in Abschnitt 2.5 vorgestellt werden. Sie umfassen alle Formalismen und ihre technische Umsetzung, die mit unscharfen Begriffen operieren können. Dazu gehören beispielsweise auch die mehrwertige Logik [104, 105, 106] (siehe Abschnitt 2.3), die Theorie rauer Mengen („rough set theory“) [125, 126], die fraktale Logik [127, 128, 129] und die Theorie weicher Mengen („soft set theory“) [115]⁶⁾.

So wie Unschärfe im praktischen Leben allgegenwärtig ist, findet sie sich auch in verschiedenen Elementen der Kundenakquisition wieder: der Kapazitätsbeschränkung für Akquiseaktionen, des Zieles der Kundenakquisition sowie der Bewertung, welche Aktionen für welche Kunden gut geeignet sind. Zu ihrer angemessenen Beschreibung und Handhabung werden in dieser Arbeit unscharfe Verfahren verwendet.

1.2 Datenbank-Marketing und AkquiSys

In der Literatur wird darauf hingewiesen, dass sich der Inhalt des Begriffs Marketing geschichtlich entwickelt hat. Über seine gegenwärtige Bedeutung gibt es keine einheitlichen Angaben. Vielmehr wird darauf verwiesen, dass der Begriff in der Praxis mit unterschiedlichen Inhalten versehen wird. So nennt z. B. [116] fünf Marketing-Begriffe, die sich nach der Art der Tätigkeit und nach dem Ausmaß, in dem die Belange des Tauschpartners (Nachfragers) zur Richtschnur des Handelns werden, unterscheiden. [141] weist darauf hin, dass die Sichtweise von Marketing als Philosophie einer marktorientierten Führung des gesamten Unternehmens,

5. Dieser Gedanke findet sich z. B. in der Überschrift eines Artikels über die Anwendung der Theorie unscharfer Mengen in der Automatisierungstechnik wieder: *Eine gute Theorie findet ihre Bestätigung in der Praxis* [27].

6. Ob und ggf. wie gut die genannten Theorien für die Modellierung von Unschärfe geeignet sind, wird unterschiedlich bewertet. So hält z. B. [29] weder mehrwertige Logiken noch die Theorien unscharfer und rauer Mengen für geeignet.

in welchem der Absatzmarkt den Ausgangspunkt aller strategischen und taktischen Planungen bildet, immer stärker wird. Diesen Gedanken hält [94] in Merksätzen fest:

- Marketing befasst sich mit den *Austauschbeziehungen* zwischen Institutionen/Personen und Personen/Institutionen.
- Marketing erstreckt sich auf die *Beeinflussung* der Austauschbeziehungen zur eigenen Zielerfüllung.
- Die Austauschbeziehungen sind um so wirkungsvoller beeinflussbar, je mehr von den *Ansprüchen* der Austauschpartner ausgegangen wird.

Den Schlüssel zur Beeinflussung der Austauschbeziehungen sieht [123] in der zielorientierten Gewinnung interner wie externer Informationen. Daraus ergeben sich von selbst die folgenden Fragen:

- Welche Informationen über den Austauschpartner sind für die Beeinflussung der Austauschbeziehungen notwendig?
- Wie können Informationen über den Austauschpartner zur Beeinflussung der Austauschbeziehungen eingesetzt werden?
- Wie können solche Informationen besorgt werden?

Alle Fragen betreffen die Beschaffung und Nutzung geeigneter Informationen zur Erlangung des Marketing-Ziels. Eine moderne Methode dafür ist das *Datenbank-Marketing* („database marketing“). Im deutschsprachigen Raum taucht der Begriff etwa seit Mitte der 80er Jahre auf [101]. Bereits 1994 setzten 62 % der Unternehmen im Versicherungsgewerbe und 55 % im Bankwesen Datenbank-Marketing ein [102]. Heutzutage sind Kundenbindungsprogramme, wie z. B. *Payback* und *Miles & More*⁷⁾ und die damit verbundene Anwendung des Datenbank-Marketings im allgemeinen Bewußtsein. Doch nur Wenige werden vermuten, dass auch Wohltätigkeitsorganisationen davon Gebrauch machen [130].

Datenbank-Marketing und Wissensentdeckung in Datenbeständen

Datenbank-Marketing wird in der Literatur relativ einheitlich definiert: [171] bestimmt Datenbank-Marketing allgemein als „*Marketingsteuerung mit Datenbankinformationen*“, [101] spricht von „*Marketing auf der Basis kundenindividueller, in einer Datenbank gespeicherter Information*“ und [143] beschreibt es als die „*Lehre von der datenbankengesteuerten Kommunikation mit Marketing-Zielgruppen*“. Von den Eigenheiten des Sozial-Marketings (das heißt nicht-kommerziellen Marketings) abgesehen, lassen sich folgende Merkmale hervorheben:

- Datenbank-Marketing basiert auf elektronisch zugänglichen Informationen über Kunden⁸⁾, die Rückschlüsse auf das Konsumverhalten⁹⁾ der Kunden zulassen.

7. siehe <http://www.payback.de>; <http://www.milesandmore.de>

8. Der Begriff Kunde schließt hier und im Folgenden die ganze Spannweite vom Stammkunden bis zu möglichen neuen Kunden ein, zu denen bisher u. U. keinerlei Verbindung bestand und die vielleicht noch nicht einmal Verwender der Produktart/Dienstleistungsart waren, die aber zumindest gute Verwendungsvoraussetzungen aufweisen, das heißt im Einzelnen: Kunden (im eigentlichen Sinne), Interessenten, Kontaktierte, Verwender, potenzielle Verwender [101].

- Datenbank-Marketing dient dem Ziel, (perspektivisch) Gewinn zu erwirtschaften¹⁰⁾.
- Datenbank-Marketing liegt die Annahme zu Grunde, dass durch geeignete Formen und Inhalte der Kommunikation (bzw. den Abbruch der Kommunikation) mit dem Kunden, das o. g. Ziel günstig beeinflusst werden kann.
- Datenbank-Marketing geht weiterhin davon aus, dass die geeigneten Formen und Inhalte der Kommunikation mit dem Kunden aus den Informationen über den Kunden abgeleitet werden können.

Die folgende Definition fasst diese Merkmale zusammen:

Definition 1.2: *Datenbank-Marketing.*

Datenbank-Marketing ist die Beschaffung elektronisch zugänglicher Informationen, die Rückschlüsse auf das Konsumverhalten von Kunden zulassen, die Auswertung dieser Informationen mit dem Zweck, Erkenntnisse über das Konsumverhalten zu gewinnen, die Umsetzung solcher Erkenntnisse in Formen und Inhalte der Kommunikation mit dem Kunden (bzw. der Abbruch des Kundenkontakts), die das Endziel der Erwirtschaftung von Gewinn günstig beeinflussen, und schließlich die Erfassung der Kundenreaktion und ggf. ihre Weiterverarbeitung. □

Aus den Merkmalen folgen die im Rahmen des Datenbank-Marketing zu lösenden Teilaufgaben¹¹⁾:

1. Informationsgewinnung (nach Kundenreaktion),
2. Kundenanalyse,
3. Identifikation von Potenzial und Risiken,
4. Kundenspezifische Marketingmaßnahmen.

Die **Aufgaben 1 und 3** liegen außerhalb der Betrachtung dieser Arbeit.

Verfahren zur Kundenanalyse (**Aufgabe 2**) gehören zum Bereich der *Datenmustererkennung* („data mining“) bzw. der *Wissensentdeckung in Datenbanken*¹²⁾ („knowledge discovery in databases“, abgekürzt: „KDD“)¹³⁾. Der Bezug zum Datenbank-Marketing ist in folgender häufig zitierten Definition augenscheinlich:

-
9. Das Konsumverhalten bezieht sich neben Sachleistungen auch auf Dienstleistungen. Man spricht dann von *Dienstleistungsmarketing*, das einige Besonderheiten aufweist [73, 111].
 10. Darin unterscheidet es sich z. B. von rein informativen Umfragen und Erhebungen (sofern es solche überhaupt gibt).
 11. [101] spricht vom *RADAR*-Modell: **R**esearch, **A**nalysis, **D**etection, **A**ction, **R**eaction.
 12. Die deutschen Begriffe *Datenmustererkennung* und *Wissensentdeckung in Datenbanken* für die üblicherweise gebräuchlichen englischen Begriffe „data mining“ und „knowledge discovery in databases“ finden sich z. B. in [24] bzw. [171].
 13. Im kommerziellen Bereich werden beide Begriffe häufig synonym gebraucht. Genau genommen bezeichnet aber Datenmustererkennung nur den Analyseschritt, in dem Hypothesen gesucht und bewertet werden [171].
-

Definition 1.3: *Wissensentdeckung in Datenbanken (kurz: WED).*

Wissensentdeckung in Datenbanken ist der nichttriviale Prozess der Identifikation gültiger, neuer, potenziell nützlicher und schlussendlich verständlicher Muster in (großen) Datenbeständen [171 nach 49]. □

Viele Kundenmerkmale und die durch Datenmustererkennung aus Kundenmerkmalen gewinnbaren Verallgemeinerungen über Kundengruppen sind unscharf. Daher kommen auch in der WED unscharfe Verfahren zur Anwendung, z. B. die unscharfe Clusteranalyse [119, 77]. Der HUBUR¹⁴)-Ansatz des Verfassers leitet Datenmuster in der Form unscharfer Regeln aus Datenbeständen ab [158, 159, 160, 161]. Da sich die Arbeit auf die unscharfe Optimierung konzentriert, würde die Darstellung unscharfer Verfahren für die Wissensentdeckung in Datenbanken zu weit führen.

Die Auswahl kundenspezifischer Marketingmaßnahmen (**Aufgabe 4**) wurde in der Vergangenheit von Marketing- oder Vertriebsmitarbeitern aufgrund ihrer Erfahrung getroffen – mit mehr oder weniger starkem subjektivistischem Einschlag. Willy Loman, Arthur Millers tragischer Handlungsreisender drückt sein Erfolgsrezept so aus:

„Because the man who makes an appearance in the business world, the man who creates personal interest, is the man who gets ahead. ...

It's not what you say, it's how you say it – because personality always wins the day.“ [113]

Offensichtlich ist es für Vertriebsmitarbeiter wichtig, einen möglichst großen Anteil der Arbeitszeit kommunikativen Tätigkeiten widmen zu können und die Zeiten für Vorbereitung und Verwaltung, für Reisen und Warten zu minimieren [143]. Aus diesem ökonomischen Zwang folgt – heute, nicht zur Zeit Willy Lomans – die Notwendigkeit des Einsatzes von Software-Werkzeugen in der Kundenakquisition. Die damit einhergehende Steigerung der Arbeitsproduktivität und Kostenreduktion ist insbesondere im kostenaufwändigen Neukundengeschäft von Bedeutung [2]. *AkquiSys* ist ein solches Software-Werkzeug. Die in der vorliegenden Arbeit entwickelten Verfahren schaffen die Grundlage zu einem echten Entscheidungsunterstützungssystem („decision support system“) zur Unterstützung der Kundenakquisition, insbesondere bei der Auswahl kundenspezifischer Marketingmaßnahmen.

***AkquiSys* und Entscheidungsunterstützung im Marketing**

AkquiSys ist ein Software-Werkzeug, mit dem eine Kampagne zur Neukundengewinnung geplant, durchgeführt und überwacht werden kann [146]. Es unterstützt Innen- und Außendienstmitarbeiter, indem es zahlreiche Routinearbeiten übernimmt, wie die Erfassung der Kundenreaktionen auf Akquiseaktionen und die Einleitung weiterer Akquiseaktionen in Abhängigkeit von der Kundenreaktion sowie die Verwaltung der Kundendaten. Eine Fachzeitschrift für Finanzdienstleistungen führt über *AkquiSys* aus:

„Dieses ... Organisationsinstrument erfasst alle Schritte im Akquisitionsprozeß mit dem Ziel eines Vertragsabschlusses.“

14. HUBUR = **H**äufigkeits**b**asierte **u**nscharfe **R**egeln-Generierung

Dabei durchlaufen die Kontaktadressen die einzelnen Aktionen des Akquiseplanes (z. B. Anschreiben, Infobriefe, Telefonate, Inserate und persönliche Kontakte) nach vorgegebenen Bedingungen und Arbeitsanweisungen, wobei der „Bearbeitungsstatus“ eines jeden Kontaktes und der Erfüllungsstand in jedem Akquiseschritt erfasst und kontrolliert werden kann. Die Registrierung der Kosten und Zeitaufwände für jede Aktion und für jeden Kontakt erlaubt eine Bewertung des Erfolgs-/Nutzens-Verhältnisses. Der alternative Vergleich unterschiedlicher Akquisepläne ermöglicht die zielgerichtete Erfolgsoptimierung der Kundenakquisition.“ [30]

Der Ablauf einer Kundenakquisition kann unter dem Gesichtspunkt des betrieblichen Aufwands durch zwei sich überlagernde Teilprozesse aufgefasst werden: einen *organisationsbezogenen* und einen *persönlichkeitsbezogenen* Anteil (siehe Abb. 1.1).

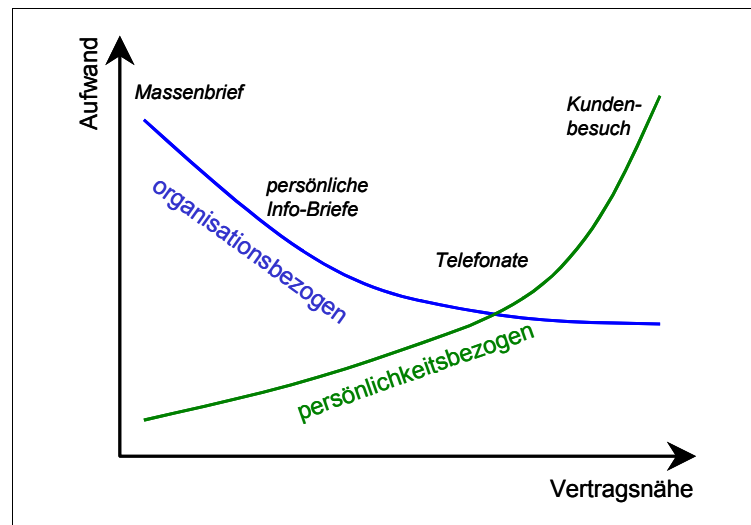


Abbildung 1.1: Zeitliche Entwicklung des Aufwands für organisationsbezogene und persönlichkeitsbezogene Teilprozesse bei der Kundenakquise. Die angegebenen Akquiseaktionen sind Beispiele.

1. Beim organisationsbezogenen Teilprozess dominiert die *Handhabung großer Datenmengen*, der Versand verschiedener Serienbriefe und Infosendungen großer Stückzahl.
2. Mit der direkten Kontaktaufnahme (Telefon, Präsentation usw.) wird der weitere Verlauf des Akquiseprozesses zunehmend durch die *Persönlichkeit* von Akquisiteur und Kunde bestimmt [146, 147].

Der Adressenstamm, mit dem ein kleines bis mittelgroßes Unternehmen die Neukundengewinnung beginnt, stammt häufig von Adressenvermittlern („list broker“). Aufbereitete Adressdaten, z. B. auf der Grundlage einer Mikrosegmentierung, können sich solche Unternehmen normalerweise nicht leisten [2, 14]. Daher beginnt die Kundenakquisition nach dem Gießkannenprinzip mit relativ wenigen Rückläufern, und unter diesen befinden sich wiederum wenige aussichtsreiche Kunden. Die Anzahl der mit solchen initialen Akquiseaktionen bearbeiteten Kunden darf dennoch nicht beliebig hoch gewählt werden, damit die aufwändigen persönlichkeitsbezogenen Akquiseaktionen zu einem späteren Zeitpunkt zeitnah zu den Kunden-Reaktionen durchgeführt werden können und kein Verzug durch Kapazitätsprobleme auftritt. Auch ist zu beachten, dass die Kosten für die – im Einzelnen eher kostengünstigen – initialen Akquiseakti-

onen insgesamt in einem wirtschaftlich vertretbaren Verhältnis zu den voraussichtlichen Erträgen stehen.

In der Phase, in der der persönlichkeitsbezogene Teilprozess den meisten Aufwand erzeugt, rückt eine weitere Frage in den Vordergrund: die Wahl der im Sinne des Akquiseerfolgs optimalen Akquiseaktion. Das Hauptziel besteht dabei in der Herstellung einer *Eins-zu-Eins-Kundenbeziehung* („one-to-one relationship with customer“) [15].

Die Lösung dieser Probleme ist mit dem Software-Werkzeug *AkquiSys* nicht direkt möglich, sondern nur durch Ausprobieren:

„Der alternative Vergleich unterschiedlicher Akquisepläne ermöglicht die zielgerichtete Erfolgsoptimierung der Kundenakquisition.“ [30]

Tatsächlich würde ein Entscheidungsunterstützungssystem („decision support system“) benötigt, dessen Funktionalität über die Organisation und Auswertung der Kundenakquisition hinausgeht. Die vorliegende Arbeit entwickelt theoretische und praktische Grundlagen dafür.

- Das wichtigste Hilfsmittel von *AkquiSys* zur Abbildung und Überwachung einer Kundenakquisition, die in obigen Zitaten erwähnten Akquisepläne, wird formalisiert und um einen *Wahrscheinlichkeitsraum* erweitert, mit dem praktisch handhabbare Verfahren zur *Simulation* des Kundenverhaltens hergeleitet werden.
- Die Kapazitäts-optimale Beschickung von Akquiseplänen, das heißt die Festlegung, wieviele Kunden wann mit der initialen Akquiseaktion bearbeitet werden sollen, wird als *unscharfes stochastisches Optimierungsproblem* formuliert, seine Lösbarkeit untersucht und praktisch durchführbare Lösungsverfahren entwickelt.
- Die Akquisepläne werden zu dynamischen Akquiseplänen erweitert. Dadurch wird eine Marketingstrategie formal zu einer Akquise-Entscheidungsregel eines *mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit*. Auf dieser Grundlage ist eine Optimierung der Akquise-Entscheidungsregel zur Auswahl der im Sinne des Akquiseerfolgs optimalen Akquiseaktionen möglich. Überdies erlaubt das Modell des zeitlichen Akquiseplans die Modellierung der im Verlauf der Akquisition wachsenden Informationen über die beworbenen Kunden, führt auf Grundlage dieser Informationen eine permanente Kundensegmentierung durch und berücksichtigt diese bei der Wahl der Akquiseaktionen.

1.3 Aufbau der Arbeit, Konventionen

Konventionen

In den nachfolgenden Kapiteln dienen folgende Konventionen der Einheitlichkeit der Darstellung:

- Definitionen und Sätze enden mit dem Zeichen: \square .
- In Definitionen werden zunächst die definierten Begriffe in kursiver Schrift angeführt, z. B.:

Definition 2.7: *t-Norm, t-Konorm, s-Norm.*

-
- Beweise enden mit dem Zeichen: ■.
 - Definitionen, Abbildungen, Tabellen und Algorithmen sind innerhalb der Kapitel jeweils fortlaufend und mit vorangestellter Kapitel-Nummer durchnummeriert.
 - Sätze und Hilfssätze sind innerhalb der Kapitel einheitlich fortlaufend und mit vorangestellter Kapitel-Nummer durchnummeriert.
 - Die Nummerierung von Korollaren erfolgt fortlaufend mit vorangestellter Nummer des Satzes, auf den sie sich beziehen.
 - Zur Vermeidung von Informationsverlusten bei der Übersetzung sind Zitate in der Originalsprache, in Deutsch oder in Englisch angegeben.
 - Zitate stehen in Anführungszeichen und sind kursiv geschrieben, z. B.:
 „Daß ich nicht mehr mit sauerem Schweiß
 Zu sagen brauche, was ich nicht weiß, ...“ [61]
 - Literaturreferenzen stehen in eckigen Klammern mit ausgeschriebenen Nachnamen des Autors und Jahreszahl des Erscheinens. Mehrere aufeinanderfolgende Literaturreferenzen sind in einem Klammernpaar zusammengefasst, z. B. [9], [176, 11].
 - Querverweise auf Kapitel, Abschnitte, Abbildungen und dergleichen werden ohne Seitennummern angegeben.
 - Querverweise auf Gleichungen enthalten in der Regel nur die Gleichungsnummer ohne vorangestelltes „Glg.“ oder „Gleichung“, z. B. (4.6).
 - Soweit möglich werden deutsche Begriffe verwendet. Zu Informationszwecken oder wenn der deutsche Begriff unüblich ist, sind auch die entsprechenden englischen Begriffe angegeben.
 - Englische Begriffe stehen in Anführungszeichen, z. B. „decision support system“.
 - Zur Hervorhebung von Wörtern wird in der Regel *Kursivschrift* und in einzelnen Fällen, zu besonders deutlicher Hervorhebung, **Fettschreibung** verwendet.
 - Eigennamen werden in der Regel in *Kursivschrift* geschrieben, z. B. *CPLEX*.
 - Der Text folgt der neuen deutschen Rechtschreibung. Ältere Zitate stehen in alter Rechtschreibung.

Aufbau der Arbeit

Kapitel 2 führt in die *Theorie unscharfer Mengen* ein, soweit dies für das Verständnis der Arbeit erforderlich ist. Es stellt im Abschnitt 2.1 die formalen Mittel für die Darstellung von *Unschärfe* bereit und beschreibt im Abschnitt 2.5 ihre Einbettung in das allgemeinere Gebiet des „Soft Computing“. Darüber hinaus hielt es der Verfasser für notwendig, im Abschnitt 2.2 auf die *Kritik* an der Theorie unscharfer Mengen einzugehen und in den Abschnitten 2.3 und 2.4 zu zeigen, dass diese Kritik keineswegs originell ist und überwiegend Diskussionen wiederholt, die bereits im Zusammenhang mit Lukasiewicz's mehrwertiger Logik geführt worden sind.

Das im Software-Werkzeug *AkquiSys* eingesetzte Hilfsmittel der *Akquisepläne* wird im Abschnitt 3.1 auf eine *mathematische Grundlage* gestellt. Danach führt Abschnitt 3.2 einen

Wahrscheinlichkeitsraum ein und zeigt, wie er ausgehend von eingängigen, vom Nutzer festzulegenden Größen praktisch berechnet werden kann. Damit ist die zeitliche *Simulation* des mit einem Akquiseplan gesteuerten stochastischen Kundenverhaltens möglich. Nach der Einführung einiger *ökonomischer Kenngrößen*, deren Simulation interessant ist, beschreibt Abschnitt 3.3 die Implementierung der Simulation als *Maple*-Programm. Anhand von *Fallbeispielen* wird das Verfahren illustriert und untersucht. Abschnitt 3.4 leitet zu auf der Simulation aufbauenden Optimierungsproblemen über.

Als erste große Optimierungseinheit behandelt Kapitel 4 die *optimale Beschickung eines Akquiseplans*. Zunächst beschreibt Abschnitt 4.1 die Behandlung der unscharfen Kapazitätsschranken, das heißt wie für das unscharfe Problem ein *scharfes Äquivalent* gewonnen werden kann. Abschnitt 4.2 behandelt drei Optimierungsprobleme mit einem Erwartungswertansatz sowie dem Verfahren der *Optimierung mit zufälligen Nebenbedingungen* („chance constrained programming“). Da beide nur bedingt geeignet sind, untersucht Abschnitt 4.3 die Struktur des Optimierungsproblems und stellt es als Variante eines *Mehrgüterfluss-Problems* dar. Für seine Modellierung wird der Akquiseplan zum *zeitlichen Akquiseplan* erweitert. In diesem Abschnitt wird auch die Erzeugung des deterministischen Äquivalents des stochastischen Problems mit Hilfe von *Szenarien* gezeigt sowie die beiden darauf aufbauenden Lösungsverfahren: *Mittelwert-Verfahren* und erweitertes *Lagrange-Verfahren*. Deren Umsetzung in ein C++-Computerprogramm unter Einsatz von *CPLEX* wird im Abschnitt 4.4 genau beschrieben. Zur Bewertung der Verfahren wird mit einem wirklichkeitsnahen *Fallbeispiel* ein umfangreiches Untersuchungsprogramm durchgeführt.

Im Kapitel 5 untersucht der Abschnitt 5.1 die Stärken und Schwächen der Kundengewinnung mit Akquiseplänen, insbesondere im Zusammenhang mit der *Kundensegmentierung*. Daraus wird über den Zwischenschritt des *erweiterten Akquiseplans* (Abschnitt 5.2) im Abschnitt 5.3 das Modell des *dynamischen Akquiseplans* entwickelt, bei dem die Akquiseaktionen nicht fest mit Kunden-Reaktionen verknüpft sind. Darauf aufbauend formuliert Abschnitt 5.4 das *unscharfe stochastische Optimierungsproblem* zur Berechnung einer Ertrags-optimalen Akquise-Entscheidungsregel.

Kapitel 6 behandelt die Lösung des im vorigen Kapitel eingeführten Optimierungsproblems, das heißt die Berechnung einer Entscheidungsregel zur Bestimmung von Akquiseaktionen, die unter bestimmten unscharfen Nebenbedingungen und bezüglich eines unscharfen Zieles optimal sind. Zunächst werden im Abschnitt 6.1 Grundbegriffe der *dynamischen Optimierung* von Bellman eingeführt. Der von Bellman und Zadeh 1970 eingeführte Begriff der *unscharfen Umgebung* wird im Abschnitt 6.2 motiviert und erklärt. Die Lösung von deterministischen und stochastischen dynamischen Optimierungsproblemen in unscharfer Umgebung mit *fester Stufenzahl* wird im Abschnitt 6.3 behandelt. Für die Bestimmung optimaler Akquiseaktionen bei der Kundenakquisition wird das Modell erweitert, so dass das *Prozessende implizit*, durch Erreichen eines Endzustandes festgelegt wird. Die hauptsächlichen theoretischen Ergebnisse werden im Abschnitt 6.4 über die *dynamische Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit* hergeleitet. Unter anderem wird gezeigt, unter welchen Bedingungen eine Akquise-Entscheidungsregel (Politik) zulässig ist und wie eine optimale Politik als Lösung eines *Fixpunktproblems* iterativ berechnet werden kann. Schließlich wird gezeigt, dass diese Hauptergebnisse für eine ganze Klasse von *t-Normen* gültig sind. Die praktische Berechnung einer optimalen Politik ist Gegenstand von Abschnitt 6.5. Im Abschnitt 6.6 wird gezeigt, wie ein dynamischer Akquiseplan für die praktische Anwendung festgelegt werden kann. Die im vorigen Abschnitt

ermittelten Verfahren werden in *Matlab*-Programme umgesetzt und mit realistischen *Fallbeispielen* illustriert und untersucht.

Abschließend gibt Kapitel 7 einen Überblick über die Arbeit und bewertet ihre Ergebnisse (Abschnitt 7.1). Mögliche *Erweiterungen*, bis hin zur Entwicklung eines Software-Prototyps, werden im Abschnitt 7.2 aufgezeigt. Hinweise auf *Anwendungsgebiete* im Abschnitt 7.3 schließen die Arbeit ab.

Neben dem Inhaltsverzeichnis (Seite V) erleichtern die folgenden Verzeichnisse die Orientierung für den Leser:

- das Abbildungsverzeichnis (Seite IX),
- das Tabellenverzeichnis (Seite XIII),
- das Verzeichnis der Definitionen (Seite XV),
- das Verzeichnis der Sätze (Seite XIX),
- das Verzeichnis der Algorithmen (Seite XXI) und
- das Literaturverzeichnis (Seite 363).

2 Unscharfe Logik

Die in dieser Arbeit in den Kapiteln 4 – 6 vorgestellten Verfahren gründen auf der unscharfen Logik. Für die Definition grundlegender Begriffe und zur Vermeidung von Missverständnissen führt dieses Kapitel zunächst ausführlich in die unscharfe Logik ein. Dabei werden die Beziehungen zu anderen, älteren Wissensgebieten berücksichtigt und auf die Kritik an der unscharfen Logik eingegangen. Der mit dieser Diskussion vertraute Leser möge ihm geläufige Passagen überschlagen.

Die unscharfe Logik bereitet in dreierlei Hinsicht Schwierigkeiten. Erstens scheint die Kombination von „unscharf“ und „Logik“ widersprüchlich: Wie kann die „wissenschaftlich strenge“ Disziplin der Logik, „unscharf“, also „ungenau“ sein? Zweitens werden mit dem Begriff unterschiedliche Inhalte verbunden – auch von Jenen, die sich mit unscharfer Logik beschäftigen. Und schließlich gibt es seit Einführung der unscharfen Logik einen nicht enden wollenden Streit über die Legitimität dieser Disziplin. Einige Wissenschaftler halten sie schlechterdings für überflüssig.

Der Name „unscharfe Logik“ ist keine Bewertung einer Logik als unscharf, sondern besagt, dass hier ein unscharfes Verfahren gemäß Def. 1.1 vorliegt, das heißt ein mathematisch sauber definiertes Verfahren, das mit unscharfen Begriffen zu operieren in der Lage ist. Grundbegriffe der unscharfen Logik werden in Abschnitt 2.1 vorgestellt. Auf Kritik an der unscharfen Logik geht Abschnitt 2.2 ein. Zur besseren Einordnung wird die unscharfe Logik in den Abschnitten 2.3 und 2.4 in Beziehung zur mehrwertigen Logik und zur Wahrscheinlichkeitsrechnung gesetzt. Neuere Weiterentwicklungen werden im Abschnitt 2.5 angerissen.

Der Streit über die Legitimität der unscharfen Logik ist eine Auseinandersetzung über die Akzeptanz von Unschärfe, über die Notwendigkeit, mit Unschärfe zu operieren und, falls letzteres bejaht wird, über das geeignete Verfahren dafür. Zur Existenz von Unschärfe und zur Notwendigkeit unscharfer Verfahren wurde bereits in Abschnitt 1.1 Stellung genommen. Ergänzend kommt der Teil von Abschnitt 2.5 hinzu, der sich mit der rechentechnischen Verarbeitung von natürlicher Sprache („computing with words“) befasst. Der Kritik an der unscharfen Logik sind die Abschnitte 2.2 und insbesondere 2.4 gewidmet.

2.1 Was ist unscharfe Logik?

Unscharfe Logik ist ein unscharfes Verfahren gemäß Def. 1.1. Ihre erfolgreiche industrielle Anwendung am Anfang der 80er Jahre¹⁾ führte zu ihrem Einsatz in zahlreichen Konsumartikeln [185], inzwischen auch in Deutschland [26, 112].

In diesem Abschnitt werden zunächst die verschiedenen Seiten der unscharfen Logik benannt und in den Definitionen 2.1 – 2.3 der Begriff der *unscharfen Menge* eingeführt. Danach werden *logische Operatoren* der unscharfen Aussagenlogik definiert, darunter in Def. 2.7 die so genannten *t-Normen*, die Verallgemeinerungen der entsprechenden logischen Operatoren der

1. Einen allgemeinverständlichen Überblick über die Entwicklung der unscharfen Logik gibt [110]. Die ersten industriellen Anwendungen der unscharfen Logik sind die Regelung eines Zementofens im Jahr 1980 [76] und die Beschleunigungs- und Bremsregelung der U-Bahn im japanischen Sendai, die 1987 ihren Betrieb aufnahm [174].

klassischen Aussagenlogik darstellen. Von ihnen werden dann *Operatoren auf unscharfen Mengen* (Def. 2.11) abgeleitet. Schließlich zeigt das *Erweiterungsprinzip* in Def. 2.12, wie „klassische“ Funktionen auf unscharfen Definitionsbereichen definiert werden können. Abschließend wird kurz auf *unscharfe* und *linguistische Variable* eingegangen, die in zahlreichen ingenieurtechnischen Anwendungen von Bedeutung sind. Weitere Begriffe werden in den Kapiteln 4 – 6 erklärt.

Heute wird zwischen *unscharfer Logik im engen Sinn* und *unscharfer Logik im weiten Sinn* [185, 189, 190] unterschieden.

- a) *Unschärfe Logik im engen Sinn* ist eine Sammlung logischer Systeme mit approximativen Schlussregeln, die als Sonderfälle zweiwertige und mehrwertige Systeme enthalten (vgl. Abschnitt 2.3).
- b) *Unschärfe Logik im weiten Sinn* ist die *Theorie unscharfer Mengen* („fuzzy set theory“). Zadeh unterscheidet vier Aspekte dieser Theorie:
 - der logische Aspekt ist identisch mit unscharfer Logik im engen Sinn;
 - der mengentheoretische Aspekt beschäftigt sich mit Mengen, deren Grenzen unscharf sind;
 - der relationale Aspekt betrachtet die Darstellung und Analyse von unscharfen Abhängigkeiten – hierzu gehören die meisten Anwendungen der unscharfen Logik;
 - der erkenntnistheoretische Aspekt ist mit dem logischen Aspekt verbunden und behandelt Wissensrepräsentation, Informationssysteme, unscharfe Datenbanken und die Möglichkeitstheorie („possibility theory“) und Wahrscheinlichkeitsrechnung.

Heutzutage ist mit „unscharfer Logik“ meistens die Theorie unscharfer Mengen gemeint. Ihre Grundlage ist der Begriff der *unscharfen Menge*, der von Zadeh 1965 in seiner Arbeit *Fuzzy Sets* eingeführt worden ist. In der Literatur finden sich verschiedene Definitionen. Zur besseren Orientierung wird zunächst ein kurzer Überblick, ohne Anspruch auf Vollständigkeit gegeben und diskutiert, und danach wird Def. 2.1 formuliert.

- „A fuzzy set (class) A in X is characterized by a membership (characteristic) function $f_A(x)$ which associates with each point in X a real number in the interval $[0, 1]$.“ [177]
- In [37] ist eine Menge A über einer Grundmenge X eine unscharfe Menge, wenn der Wertebereich ihrer charakteristischen Funktion μ_A das ganze reelle Intervall $[0, 1]$ sein kann. Weiterhin wird ausgeführt:

„ A is completely characterized by the set of pairs

$$A = \{(x, \mu_A(x)), x \in X\}. \quad (2.1)$$

A more convenient notation was proposed by Zadeh When X is a finite set $\{x_1, \dots, x_n\}$, a fuzzy set on X is expressed as

$$A = \mu_A(x_1)/x_1 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_A(x_i)/x_i. \quad (2.2)$$

When X is not finite, we write

$$A = \int_X \mu_A(x_i) / x_i. \text{“}^{(2)} \text{“} \quad (2.3)$$

- „If X is a collection of objects denoted generically by x then a fuzzy set \tilde{A} in X is a set of ordered pairs:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

... Elements with a zero degree of membership are normally not listed. “⁽³⁾ [194]

- In [5] wird eine unscharfe Menge durch ihre Zugehörigkeitsfunktion „charakterisiert“.
- „A fuzzy set A over some universe of discourse X (which itself is a classical set) is characterized by its membership function $\mu_A : X \rightarrow [0, 1]$... The fuzzy sets will be identified with their membership functions ... Fuzzy sets are thus generalized characteristic functions.“ [65]
- „Sei G eine (stets klassische) Grundmenge, μ_A eine (stets klassische) Funktion der Grundmenge G in das Einheitsintervall $[0, 1]$ der reellen Achse:

$$\mu_A : G \rightarrow [0, 1].$$

Dann heißt die Menge A aller Paare $(x, \mu_A(x))$

$$A = \{(x, \mu_A(x)) \mid x \in G\}$$

eine Fuzzy-Menge (unscharfe Menge, fuzzy set) über G .“ [25]

- “... Such a function is called a membership function, and the set defined by it a fuzzy set. ... Two distinct notations are most commonly employed in the literature to denote membership functions. In one of them, the membership function of a fuzzy set A is denoted by μ_A ; that is

$$\mu_A : X \rightarrow [0, 1].$$

... In the other, the function is denoted by A and has, of course the same form:

$$A : X \rightarrow [0, 1]. \text{“} [93] \quad (2.4)$$

- In [21] wird eine unscharfe Menge durch eine Funktion μ von einer Grundmenge Ω in das reelle Einheitsintervall $\mu : \Omega \rightarrow [0, 1]$ „charakterisiert“.

-
2. Pluszeichen und Bruchstrich werden hier mit anderer Bedeutung als in der Arithmetik gebraucht: Der Bruchstrich vereint jedes Element mit seinem Zugehörigkeitswert zu einem Wertepaar (in (2.1) durch ein rundes Klammernpaar symbolisiert). Die Vereinigung dieser Wertepaare zur unscharfen Menge wird durch Pluszeichen bezeichnet (in (2.1) durch ein geschweiftes Klammernpaar symbolisiert). In übertragener Bedeutung werden Summenzeichen und Integralzeichen wie das Pluszeichen gebraucht.
 3. Die von Zimmermann verwendete Tilde zur Unterscheidung unscharfer Mengen von „klassischen“ Mengen ist in der unscharfen Logik nicht allgemein gebräuchlich.
-

- „Sei X eine Grundmenge. Eine unscharfe Menge (Untermenge) über X ist ein Denotat (z. B. „klein“, „groß“, „A“), das durch eine sogenannte „Zugehörigkeitsfunktion“ (z. B. $\mu_{\text{klein}}, \mu_{\text{groß}}, \mu_A$) spezifiziert wird.“ [150]

Zusammenfassend können obige Definitionen in zwei Gruppen eingeteilt werden:

Zur ersten Gruppe gehören [177, 5, 65, 93, 21, 150]. Für sie ist eine unscharfe Menge durch eine verallgemeinerte charakteristische Funktion, eine so genannte Zugehörigkeitsfunktion „charakterisiert“ bzw. „spezifiziert“. Dabei wird in den meisten Fällen gar nicht oder nur undeutlich gesagt, was durch die Zugehörigkeitsfunktion charakterisiert wird. Formulierungen wie „über einer Grundmenge“ bzw. „over a universe of discourse“ deuten an, dass die Charakterisierung den Elementen einer „klassischen“ Grundmenge, also einer „klassischen“ Teilmenge gilt⁴⁾. Eine Ausnahme ist [150], wo die unscharfe Menge ein „Denotat“, also eine „Bezeichnung“ ist, die die Zugehörigkeitsfunktion charakterisiert. Ein „Extremfall“ ist die Definition in [65], die die unscharfe Menge mit der Zugehörigkeitsfunktion gleichsetzt, wie es auch (2.4) in [93] andeutet.

Zur zweiten Gruppe gehören [194, 25]. Sie definieren eine unscharfe Menge durch Aufzählung von Elementen der „klassischen“ Grundmenge und deren Zugehörigkeitswerten zu der zu definierenden unscharfen Menge, wobei in der Regel nur solche Elemente der Grundmenge aufgeführt werden, deren Zugehörigkeitswert größer als null ist⁴⁾. Einen Zusammenhang zwischen beiden Gruppen deutet [37] durch die Bemerkung an, dass eine unscharfe Menge durch die Aufzählung der Element-Zugehörigkeitswert-Paare (2.1) – (2.3) vollständig charakterisiert wird.

Genauer gilt: Durch die Aufzählung von Element-Zugehörigkeitswert-Paaren (x, y) mit x aus einer „klassischen“ Menge X und eindeutigem $y \in [0, 1]$ kann eine charakteristische Funktion definiert werden:

$$\mu : \begin{cases} X \rightarrow [0, 1] \\ x \rightarrow \mu(x) \end{cases}, \quad (2.5)$$

wobei unausgesprochen angenommen wird, dass für alle $x \in X$, die in der Aufzählung der Paare (x, y) nicht vorkommen, der Zugehörigkeitswert gleich null ist. Andererseits wird durch eine Zugehörigkeitsfunktion der Gestalt (2.5) eine Menge von Paaren $(x, \mu(x))$ definiert, nämlich der Graph der Funktion μ . Somit ist für eine gegebene Grundmenge X die Abbildung der Mengen mit Element-Zugehörigkeitswert-Paaren auf die Menge der Zugehörigkeitsfunktionen eine Bijektion. Dies entspricht der Darstellung von [37]. Daraus folgt, dass zur Definition einer unscharfen Menge neben der Angabe der Grundmenge entweder die Angabe der Zugehörigkeitswerte der Elemente der Grundmenge zur unscharfen Menge oder die Angabe der Zugehörigkeitsfunktion gehört. Eine darauf beruhende Definition sagt noch nichts darüber aus, was eine unscharfe Menge ist. Auch die Definition einer unscharfen Menge als „Denotat“ (also „Bezeichnung“) in [150] ist unbefriedigend, wirft sie doch die Frage auf:

4. Diese Teilmenge wird in Def. 2.4 als „Träger“ der unscharfen Menge eingeführt.

„Bezeichnung wofür?“. Zur Lösung dieses Problems sei trotz der bekannten Probleme⁵⁾ auf die „naive“ Mengendefinition von Cantor zurückgegriffen:

„Unter einer 'Menge' verstehen wir jede Zusammenfassung M von bestimmten, wohlunterschiedenen Objekten m unserer Anschauung oder unseres Denkens (welche die 'Elemente' von M genannt werden) zu einem Ganzen.“ [zitiert nach 35]

Aus obigen Überlegungen ergibt sich folgende umgangssprachliche Verallgemeinerung, die in Anlehnung an Cantor als „naive Definition“ einer unscharfen Menge bezeichnet wird.

Definition 2.1 („Naive Definition“): *Unscharfe Menge, Zugehörigkeitswert, Grundmenge.*

Eine *unscharfe Menge* („fuzzy set“) A ist die Zusammenfassung von bestimmten wohlunterschiedenen Objekten x unserer Anschauung oder unseres Denkens (welche die Elemente von A genannt werden) zu einem Ganzen, wobei bezüglich der Zugehörigkeit der Elemente zum Ganzen Abstufungen zwischen „nicht zugehörig“ und „gänzlich zugehörig“ möglich sind.

Eine unscharfe Menge A ist eindeutig bestimmt durch Angabe ihrer Elemente und für jedes Element x des Grades seiner Zugehörigkeit zu A , dem *Zugehörigkeitswert*, der zwischen null („nicht zugehörig“) und eins („gänzlich zugehörig“) liegt. Dafür gibt es zwei gleichwertige Darstellungsformen, die beide davon ausgehen, dass die Elemente von A vollständig, das heißt mit Zugehörigkeitswert eins in einer *Grundmenge* („universe of discourse“) X enthalten sind, was durch die Formulierung „unscharfe Menge A über der Grundmenge X “ angedeutet wird. \square

Die beiden Darstellungsformen genügen den formalen Ansprüchen einer mathematischen Definition.

Definition 2.2: *Unscharfe Menge (1. Darstellungsform), Grundmenge, Zugehörigkeitswert.*

1. Darstellungsform: Eine *unscharfe Menge* A über der *Grundmenge* X ist eine Menge von Paaren (x, y) ,

$$A = \{(x, y) \mid x \in X, y \in [0, 1]\},$$

wobei zu jedem $x \in X$ der *Zugehörigkeitswert* $y \in [0, 1]$ eindeutig bestimmt ist. Paare mit $y = 0$ werden üblicherweise weggelassen. X wird als Menge im Sinne der axiomatischen Mengenlehre vorausgesetzt. \square

5. Gemeint sind die Paradoxien der „naiven“ Mengenlehre, die zur axiomatischen Mengenlehre führen.

Definition 2.3: *Unschärfe Menge (2. Darstellungsform), Grundmenge, Zugehörigkeitsfunktion, Zugehörigkeitswert, Menge der unscharfen Mengen.*

2. Darstellungsform: Eine *unscharfe Menge* A über der *Grundmenge* X ist gegeben durch die *Zugehörigkeitsfunktion*

$$\mu_A : \begin{cases} X \rightarrow [0, 1] \\ x \rightarrow \mu_A(x) \end{cases},$$

die jedem Element $x \in X$ einen *Zugehörigkeitswert* $\mu_A(x)$ eindeutig zuordnet. X wird dabei als Menge im Sinne der axiomatischen Mengenlehre vorausgesetzt.

$\text{IF}(X)$ bezeichnet die *Menge der unscharfen Mengen* über X . \square

Aus der Definition folgt, dass jede scharfe Menge $A \subseteq X$ mit der charakteristischen Funktion χ_A als unscharfe Menge darstellbar ist, wenn die entsprechende Zugehörigkeitsfunktion μ_A mit der charakteristischen Funktion gleichgesetzt wird.

Durch Ersetzen von scharfen durch unscharfe Mengen läßt sich jede Theorie, die auf scharfen Mengen beruht, abstumpfen⁶⁾ („fuzzifizieren“). Auf diese Weise entstehen – neben der unscharfen Logik im engen Sinn – weitere Zweige der Theorie unscharfer Mengen, z. B. unscharfe Arithmetik, unscharfe Optimierung (vgl. Kapitel 4 und 6), unscharfe Topologie, unscharfe Graphentheorie und unscharfe Datenanalyse [185]. Die Abstumpfung beruht auf dem Erweiterungsprinzip, das in Def. 2.12 eingeführt wird.

In den folgenden Kapiteln werden die Begriffe des *Trägers* und des α -*Schnittes* einer unscharfen Menge sowie der Begriff einer *normalen unscharfen Menge* gebraucht. Sie werden nachfolgend eingeführt. In Ermangelung einer einheitlichen Schreibweise, orientiert sich der Verfasser an der Notation in [5] und [65].

Definition 2.4: *Träger.*

Für $A \in \text{IF}(X)$ wird die scharfe Menge

$$\text{supp } A = \{x \in X \mid \mu_A(x) > 0\}$$

als *Träger* („support“) der unscharfen Menge A bezeichnet. \square

Anmerkung

Eine unscharfe Menge A über der Grundmenge X wird *leer* genannt, wenn für alle $x \in X$ gilt: $\mu_A(x) = 0$. In diesem Fall ist der Träger der unscharfen Menge A die leere Menge: $\text{supp } A = \emptyset$.

6. Der deutsche Begriff „abstumpfen“ für „to fuzzify“ hat sich zwar nicht allgemein durchgesetzt, findet aber Verwendung in der Literatur (z. B. in [64]) und wird um der einheitlichen Deutschschreibung willen hier verwendet.

Definition 2.5: α -Schnitt.

Für $A \in \mathcal{IF}(X)$ wird die scharfe Menge

$${}^{\alpha}A = \{x \in X \mid \mu_A(x) \geq \alpha\}$$

als α -Schnitt („ α -cut“) bezeichnet. □

Definition 2.6: Normale unscharfe Menge.

Die unscharfe Menge $A \in \mathcal{IF}(X)$ heißt *normal*, wenn gilt:

$$\sup_{x \in X} \mu_A(x) = 1.$$

□

In der klassischen Mengenlehre werden Mengenoperatoren, wie Durchschnittbildung und Vereinigung mit Hilfe von Aussagen über ihre Elemente und logische Verknüpfungen dieser Aussagen definiert. Diese Vorgehensweise kann auf die Theorie unscharfer Mengen übertragen werden. Dazu sind zunächst logische Operatoren für unscharfe Aussagen einzuführen. Das sind Aussagen, deren Wahrheitswert alle Werte im Intervall $[0, 1]$ annehmen kann, also insbesondere nicht nur die Wahrheitswerte Null und Eins der klassischen Aussagenlogik. Mit ihrer Hilfe werden weiter unten die Operatoren auf unscharfen Mengen definiert. Viele Autoren beschränken sich bei logischen Operatoren zur Verknüpfung zweier unscharfer Aussagen auf die Minimum- und Maximum-Bildung. Andere, wie [12], axiomatisieren „unmittelbar einsichtige“ Eigenschaften solcher Operatoren und zeigen, dass nur die Minimum- und Maximum-Bildung ihnen genügen können. Der Verfasser folgt dem heute üblichen Vorgehen und führt „verallgemeinerte“ logische Und- und Oder-Operatoren ein, die so genannten *t-Normen* und *s-Normen*.

Definition 2.7: *t-Norm*, *t-Konorm*, *s-Norm*.

(a) Ein zweistelliger Operator \mathbf{t} im reellen Einheitsintervall $[0, 1]$ ist eine *t-Norm* genau dann, wenn er

(T1) assoziativ und kommutativ ist;

(T2) im ersten Argument (und damit in beiden Argumenten) schwach monoton wächst;

(T3) die Eins als neutrales Element besitzt, das heißt $u \mathbf{t} 1 = u$ für jedes $u \in [0, 1]$.

(b) Ein zweistelliger Operator \mathbf{s} im reellen Einheitsintervall $[0, 1]$ ist eine *t-Konorm* oder *s-Norm* genau dann, wenn er

(S1) assoziativ und kommutativ ist;

(S2) in jedem Argument schwach monoton abnimmt;

(S3) die Null als neutrales Element besitzt, das heißt $u \mathbf{s} 0 = u$ für jedes $u \in [0, 1]$. □

Für t-Normen und t-Konormen ist auch die Präfix-Schreibweise gebräuchlich, also z. B. $t(u, v)$ statt $u \mathbin{t} v$. Für die mehrfache Hintereinanderausführung einer t-Norm oder t-Konorm wird das Operatorsymbol mit einem Index versehen, also:

$$t_{i=1}^n(u_i) = u_1 \mathbin{t} u_2 \mathbin{t} u_3 \dots u_{n-1} \mathbin{t} u_n := u_1 \mathbin{t} (u_2 \mathbin{t} (u_3 \dots (u_{n-1} \mathbin{t} u_n) \dots))$$

für die t-Norm \mathbin{t} und analog für t-Konormen.

Den Zusammenhang zwischen beiden Klassen von Operatoren stellt eine *Negationsfunktion* her (siehe Satz 2.1), die auch für die Definition der Negation unscharfer Aussagen und des Komplements einer unscharfen Menge benötigt wird.

Definition 2.8: *Negationsfunktion, strenge Negationsfunktion, involutorische Negationsfunktion.*

Ein einstelliger Operator \mathbf{n} im reellen Einheitsintervall $[0, 1]$ ist eine *Negationsfunktion*, genau dann, wenn gilt:

$$(N1) \mathbf{n}(0) = 1, \mathbf{n}(1) = 0;$$

$$(N2) \mathbf{n} \text{ ist schwach monoton abnehmend;}$$

\mathbf{n} ist eine *strenge Negationsfunktion*, wenn außerdem gilt:

$$(N3) \mathbf{n} \text{ ist streng monoton abnehmend und stetig;}$$

\mathbf{n} ist eine *involutorische Negationsfunktion* genau dann, wenn \mathbf{n} eine strenge Negationsfunktion ist und folgende Eigenschaft besitzt:

$$(N4) \mathbf{n}(\mathbf{n}(u)) = u \text{ für alle } u \in [0, 1].$$

□

Anmerkung

Wegen der strengen Monotonieeigenschaft besitzt jede *strenge Negationsfunktion* \mathbf{n} eine Inverse \mathbf{n}^{-1} .

Satz 2.1: Gegeben sei eine strenge Negationsfunktion \mathbf{n} , die nicht notwendigerweise involutorisch sein muss. Dann ist für jede t-Norm \mathbf{t} durch

$$\mathbf{s}_{\mathbf{n}, \mathbf{t}}(u, v) := \mathbf{n}^{-1}(\mathbf{t}(\mathbf{n}(u), \mathbf{n}(v))) \quad (2.6)$$

eine t-Konorm definiert; und für jede t-Konorm \mathbf{s} ist durch

$$\mathbf{t}_{\mathbf{n}, \mathbf{s}}(u, v) := \mathbf{n}^{-1}(\mathbf{s}(\mathbf{n}(u), \mathbf{n}(v))) \quad (2.7)$$

eine t-Norm definiert.

Ist \mathbf{n} eine involutorische Negationsfunktion, dann besteht zwischen (2.6) und (2.7) die Beziehung:

$$t_{n, s_t} = t \text{ und } s_{n, t_s} = s .$$

□

Beweis: siehe [65]

Anmerkungen

1. Im Folgenden wird ausschließlich die involutorische Negationsfunktion

$$n(u) := 1 - u \quad (2.8)$$

betrachtet. Bezüglich dieser Negationsfunktion und einer gegebenen t-Norm t wird statt $s_{n, t}$ nur s_t geschrieben.

2. Geläufige t-Normen sind die Minimum- und die Produktbildung, $\min(u, v)$ bzw. $u \cdot v$ für $u, v \in [0, 1]$. Mit der Negationsfunktion (2.8) sind die entsprechenden t-Konormen (s-Normen): $\max(u, v)$ bzw. $u + v - u \cdot v$.
3. Weitere Beispiele für t-Normen finden sich in Abschnitt 6.6.3.1.

Der Implikationsoperator der unscharfen Logik bzw. die Enthaltenseinsrelation zwischen unscharfen Mengen wird mit Hilfe so genannter Φ -Operatoren eingeführt. Anmerkungen zur Motivation dieses Operators und zu den Unterschieden gegenüber dem Implikationsoperator der klassischen Aussagenlogik erfordern die Einführung weiterer Begrifflichkeiten und werden daher im Anschluss an Def. 2.10 gegeben.

Definition 2.9: Φ -Operator.

Ein zweistelliger Operator ϕ im reellen Einheitsintervall $[0, 1]$ wird als Φ -Operator (bezüglich einer gegebenen t-Norm t) bezeichnet genau dann, wenn für jedes $u \in [0, 1]$ gilt:

$$(\Phi 1) \quad v \leq w \Rightarrow u \phi v \leq u \phi w ;$$

$$(\Phi 2) \quad u t (u \phi v) \leq v ;$$

$$(\Phi 3) \quad v \leq u \phi (u t v) .$$

□

Anmerkungen

1. Die Abhängigkeit des Φ -Operators ϕ von der t-Norm t wird durch ϕ_t ausgedrückt.
2. Damit für eine gegebene t-Norm t ein Φ -Operator existiert, muss die t-Norm linksstetig sein.

Satz 2.2: (a) Für eine t-Norm t gibt es einen Φ -Operator entsprechend Def. 2.9 genau dann, wenn t linksstetig ist.

(b) Zu jeder linksstetigen t-Norm t gibt es genau einen Φ -Operator ϕ_t . Für ihn gilt

$$u \phi_t v = \max\{w \mid (u \mathbin{\text{t}} w) \leq v\} .$$

□

Beweis: siehe [65]

Im Weiteren wird von einer mehrwertigen Logik L ausgegangen, deren Wahrheitswerte im Einheitsintervall $[0, 1]$ liegen [66]. Diese Logik wird um Symbole für unscharfe Mengen ergänzt und um ein zweistelliges Prädikat ε für die Elementbeziehung zwischen Elementen einer Grundmenge X und unscharfen Mengen aus $\text{IF}(X)$. Für jeden in der so konstituierten Logik wohlformulierten Ausdruck H bezeichne $\llbracket H \rrbracket$ dessen Wahrheitswert. Für alle unscharfen Mengen $A \in \text{IF}(X)$ und alle $x \in X$ stellt die grundlegende Gleichung

$$\llbracket x \varepsilon A \rrbracket := \mu_A(x) \quad (2.9)$$

den Zusammenhang zwischen dem Wahrheitswert von $x \varepsilon A$ und dem Zugehörigkeitswert $\mu_A(x)$ her. Für alle Grundmengen X , alle Aussageformen $H(x)$, $x \in X$ der Logik L und alle unscharfen Mengen $A \in \text{IF}(X)$ bezeichne

$$A = \{x \in X \parallel H(x)\}^7)$$

den Zusammenhang

$$\mu_A(x) = \llbracket H(x) \rrbracket ,$$

das heißt:

$$A = \{x \in X \parallel H(x)\} \Leftrightarrow \mu_A(x) = \llbracket H(x) \rrbracket \text{ für alle } x \in X \text{ und alle } A \in \text{IF}(X) . \quad (2.10)$$

Weiterhin gelte

$$\llbracket \forall x H(x) \rrbracket := \inf_{x \in X} \llbracket H(x) \rrbracket \text{ und } \llbracket \exists x H(x) \rrbracket := \sup_{x \in X} \llbracket H(x) \rrbracket \quad (2.11)$$

für alle Grundmengen X und alle Aussageformen $H(x)$, $x \in X$ der Logik L [5]. Nun können unscharfe Verknüpfungsoperatoren der Logik L eingeführt werden, mit deren Hilfe – wie weiter unten gezeigt wird – die Operatoren auf unscharfen Mengen definiert werden.

Definition 2.10: *Unschärfe Konjunktion, unscharfe Disjunktion, unscharfe Implikation, unscharfe Negation.*

Für die mehrwertige Logik L und für jede linksstetige t -Norm $\mathbin{\text{t}}$ bezeichnet $\mathbin{\text{t}}$ den *unscharfen Konjunktionsoperator*, dessen Wahrheitsfunktion diese t -Norm $\mathbin{\text{t}}$

7. Man beachte den doppelten Strich in $\{x \in X \parallel H(x)\}$, der darauf hinweist, dass es sich um eine unscharfe Menge handelt. Bei scharfen Mengen wird üblicherweise nur ein Strich verwendet, z. B. $\{x \in X \mid H(x)\}$.

ist, $\vee_{\mathbf{t}}$ bezeichnet den *unscharfen Disjunktionsoperator* mit der Wahrheitsfunktion $\mathbf{s}_{\mathbf{t}}$, $\rightarrow_{\mathbf{t}}$ bezeichnet den *unscharfen Implikationsoperator*, dessen Wahrheitsfunktion der Φ -Operator $\phi_{\mathbf{t}}$ ist, und \neg bezeichnet die *unscharfe Negation*, deren Wahrheitsfunktion die Negationsfunktion \mathbf{n} ist. \square

Das heißt, für beliebige wohlformulierte Ausdrücke H_1 , H_2 und H der Logik L sowie eine gegebene t -Norm gilt:

$$\begin{aligned} \llbracket H_1 \wedge_{\mathbf{t}} H_2 \rrbracket &= \llbracket H_1 \rrbracket \mathbf{t} \llbracket H_2 \rrbracket, \\ \llbracket H_1 \vee_{\mathbf{t}} H_2 \rrbracket &= \llbracket H_1 \rrbracket \mathbf{s}_{\mathbf{t}} \llbracket H_2 \rrbracket, \\ \llbracket H_1 \rightarrow_{\mathbf{t}} H_2 \rrbracket &= \llbracket H_1 \rrbracket \phi_{\mathbf{t}} \llbracket H_2 \rrbracket, \\ \llbracket \neg H \rrbracket &= \mathbf{n} \llbracket H \rrbracket. \end{aligned} \tag{2.12}$$

Anmerkungen

1. Def. 2.10 wirft die Frage auf, weshalb der unscharfe Implikationsoperator nicht wie in der klassischen Aussagenlogik über die t -Norm und die Negationsfunktion definiert ist, z. B.

$$\llbracket H_1 \rightarrow_{\mathbf{t}} H_2 \rrbracket := (\mathbf{n} \llbracket H_1 \rrbracket) \mathbf{t} \llbracket H_2 \rrbracket, \tag{2.13}$$

sondern die Einführung des Φ -Operators $\phi_{\mathbf{t}}$ erfordert. Die Antwort liegt in folgendem Unterschied: Während es in der klassischen Aussagenlogik mehrere unterschiedliche Definitionen des Implikationsoperators gibt, führt ihre Übertragung in die unscharfe Logik zu unscharfen Implikationsoperatoren mit unterschiedlichen Eigenschaften. Die aus den unterschiedlichen Definitionen hervorgehenden Implikationsoperatoren stimmen nur für die Werte Null und Eins überein. Die Definition des unscharfen Implikationsoperators hängt letztlich von seinen gewünschten Eigenschaften ab. Eine gute Darstellung dieser Problematik findet sich in [93] und auch in [25] und [38]. Mit der Definition des unscharfen Implikationsoperators gemäß (2.13) ist für beliebiges $\llbracket H \rrbracket \in [0, 1]$

$$\llbracket H \rightarrow_{\mathbf{t}} H \rrbracket \leq \llbracket H \rrbracket,$$

während mit (2.12)

$$\llbracket H \rightarrow_{\mathbf{t}} H \rrbracket = 1$$

die typischerweise erwartete Eigenschaft folgt.

2. Die Wahl der Eigenschaften war auch bei der Definition der Disjunktion in Lukasiewiczs dreiwertiger Logik ausschlaggebend. Vergleiche dazu die Kritik von Zawirski an Lukasiewicz auf Seite 36 und in [192].
3. Mit Eigenschaft $(\Phi 2)$ aus Def. 2.9, $u \mathbf{t} (u \phi v) \leq v$ folgt

$$\llbracket H_1 \rrbracket \mathbf{t} \llbracket H_1 \rightarrow_{\mathbf{t}} H_2 \rrbracket \leq \llbracket H_2 \rrbracket.$$

Die unscharfe Implikation gibt also vor, wie stark der Modus Ponens wirkt. So folgt aus Satz 2.2 mit $\llbracket H_1 \rrbracket = 1$ die Eigenschaft $\llbracket H_1 \rrbracket \mathbf{t} \llbracket H_1 \rightarrow_{\mathbf{t}} H_2 \rrbracket = \llbracket H_2 \rrbracket$.

Mit Hilfe der unscharfen Verknüpfungsoperatoren aus Def. 2.10 können die Operatoren für unscharfe Mengen definiert werden.

Definition 2.11: *Unscharfer Durchschnitt, unscharfe Vereinigung, unscharfes Komplement, unscharfes kartesisches Produkt.*

Für jede t-Norm \mathbf{t} , die Negationsfunktion $\mathbf{n}(u) = 1 - u$ und die zugehörige t-Konorm $\mathbf{s}_{\mathbf{t}}$ sind für zwei beliebige unscharfe Mengen $A, B \in \mathbb{IF}(X)$ der *unscharfe Durchschnitt*, die *unscharfe Vereinigung*, das *unscharfe Komplement* und das *unscharfe kartesische Produkt*:

$$A \cap_{\mathbf{t}} B := \{x \in X \mid (x \varepsilon A) \wedge_{\mathbf{t}} (x \varepsilon B)\} ,$$

$$A \cup_{\mathbf{t}} B := \{x \in X \mid (x \varepsilon A) \vee_{\mathbf{t}} (x \varepsilon B)\} ,$$

$$\bar{A} := \{x \in X \mid \neg(x \varepsilon A)\} ,$$

$$A \times_{\mathbf{t}} B := \{(x, y) \mid (x \varepsilon A) \wedge_{\mathbf{t}} (y \varepsilon B)\} .$$

□

Satz 2.3: Gegeben seien die unscharfen Mengen $A, B, C, D, E \in \mathbb{IF}(X)$ mit $C = A \cap_{\mathbf{t}} B$, $D = A \cup_{\mathbf{t}} B$ und $E = \bar{A}$ bezüglich der in Def. 2.11 gegebenen Durchschnitts-, Vereinigungs- und Komplementbildung. Für die Berechnung des Zugehörigkeitswertes eines beliebigen $x \in X$ für die Mengen C , D und E gilt:

$$\mu_C(x) = \mu_A(x) \mathbf{t} \mu_B(x) ,$$

$$\mu_D(x) = \mu_A(x) \mathbf{s}_{\mathbf{t}} \mu_B(x) ,$$

$$\mu_E(x) = 1 - \mu_A(x) .$$

Beweis: Nach (2.9) und (2.10) ist $\mu_C(x) = \mu_{A \cap_{\mathbf{t}} B}(x) = \llbracket (x \varepsilon A) \wedge_{\mathbf{t}} (x \varepsilon B) \rrbracket$. Nach Def. 2.10 ist $\llbracket (x \varepsilon A) \wedge_{\mathbf{t}} (x \varepsilon B) \rrbracket = \llbracket x \varepsilon A \rrbracket \mathbf{t} \llbracket x \varepsilon B \rrbracket$. Mit (2.9) ist $\llbracket x \varepsilon A \rrbracket = \mu_A(x)$ und $\llbracket x \varepsilon B \rrbracket = \mu_B(x)$ und damit $\mu_C(x) = \mu_A(x) \mathbf{t} \mu_B(x)$.

Für $\mu_D(x)$ ist der Beweis völlig analog:

$$\mu_D(x) = \mu_{A \cup_{\mathbf{t}} B}(x) = \llbracket (x \varepsilon A) \vee_{\mathbf{t}} (x \varepsilon B) \rrbracket = \llbracket x \varepsilon A \rrbracket \mathbf{s}_{\mathbf{t}} \llbracket x \varepsilon B \rrbracket = \mu_A(x) \mathbf{s}_{\mathbf{t}} \mu_B(x) .$$

Schließlich ist $\mu_E(x) = \mu_{\bar{A}}(x) = \llbracket \neg(x \varepsilon A) \rrbracket = \mathbf{n}(\llbracket x \varepsilon A \rrbracket)$ und mit (2.8) folgt $\mathbf{n}(\llbracket x \varepsilon A \rrbracket) = 1 - \llbracket x \varepsilon A \rrbracket = 1 - \mu_A(x)$. ■

Jetzt kann das *Erweiterungsprinzip* formuliert werden, mit dem Funktionen auf scharfen Mengen auf unscharfe Mengen erweitert, also *abgestumpft* werden können.

Definition 2.12: *Erweiterungsprinzip.*

Eine Funktion $g : X^n \rightarrow Y$ wird abgestumpft, das heißt bezüglich einer gegebenen t -Norm t zu einer Funktion $g_t : [IF(X)]^n \rightarrow IF(Y)$ erweitert, indem für alle $A_1, \dots, A_n \in IF(X)$

$$B = g_t(A_1, \dots, A_n) \in IF(Y) \text{ mit}$$

$$B = \{y \in Y \parallel \exists(x_1, \dots, x_n)[((x_1 \varepsilon A_1) \wedge_t \dots \wedge_t (x_n \varepsilon A_n)) \wedge_t y = g(x_1, \dots, x_n)]\}$$

gesetzt wird. □

Mit der Bezeichnungsweise aus Def. 2.12 gilt:

Satz 2.4: Gegeben sei die scharfe Funktion $g : X^n \rightarrow Y$ und ihre unscharfe Erweiterung gemäß Def. 2.12, $g_t : [IF(X)]^n \rightarrow IF(Y)$ und $B = g_t(A_1, \dots, A_n)$. Dann gilt für die Zugehörigkeitswerte von B :

$$\mu_B(y) = \max\left\{\sup_{\substack{x_1, \dots, x_n \in X \\ y = g(x_1, \dots, x_n)}} [\mu_{A_1}(x_1) t \dots t \mu_{A_n}(x_n)], 0\right\}$$

für alle $y \in Y$.

Beweis: Nach Def. 2.12 ist

$$B = \{y \in Y \parallel \exists(x_1, \dots, x_n)[((x_1 \varepsilon A_1) \wedge_t \dots \wedge_t (x_n \varepsilon A_n)) \wedge_t y = g(x_1, \dots, x_n)]\},$$

woraus mit (2.10) und (2.11) folgt:

$$\begin{aligned} \mu_B(y) &= \sup_{x_1, \dots, x_n \in X} \llbracket ((x_1 \varepsilon A_1) \wedge_t \dots \wedge_t (x_n \varepsilon A_n)) \wedge_t y = g(x_1, \dots, x_n) \rrbracket \\ &= \sup_{x_1, \dots, x_n \in X} \llbracket (x_1 \varepsilon A_1) \wedge_t \dots \wedge_t (x_n \varepsilon A_n) \rrbracket t \llbracket y = g(x_1, \dots, x_n) \rrbracket. \end{aligned}$$

Es ist aber

$$\llbracket y = g(x_1, \dots, x_n) \rrbracket = \begin{cases} 1 & \text{für } y = g(x_1, \dots, x_n) \\ 0 & \text{für } y \neq g(x_1, \dots, x_n) \end{cases}.$$

Mit Def. 2.7 folgt aus (T3), dass $0 t 1 = 0$, und damit aus (T2), dass für alle $u \in [0, 1]$ gilt: $0 t u = 0$. Mit der üblichen Definition $\sup \emptyset = -\infty$ ist damit:

$$\begin{aligned}\mu_B(y) &= \sup_{x_1, \dots, x_n \in X} [\llbracket (x_1 \in A_1) \wedge \mathbf{t} \dots \wedge \mathbf{t} (x_n \in A_n) \rrbracket \mathbf{t} \llbracket y = g(x_1, \dots, x_n) \rrbracket] \\ &= \max \left\{ \sup_{\substack{x_1, \dots, x_n \in X \\ y = g(x_1, \dots, x_n)}} [\mu_{A_1}(x_1) \mathbf{t} \dots \mathbf{t} \mu_{A_n}(x_n)], 0 \right\}\end{aligned}$$

■

Abschließend sei kurz auf *unscharfe* und *linguistische Variable* eingegangen. Sie sind in zahlreichen ingenieurtechnischen Anwendungen von Bedeutung, insbesondere in der Regelungstechnik. Als Beispiel sei hier eine automatische Fahrgeschwindigkeitssteuerung angeführt [50]. Die Aufgabe dieser Regelung besteht in der Bestimmung der Sollbeschleunigung eines Personenkraftwagens, das heißt im graduellen Beschleunigen oder Abbremsen bzw. dem Halten der aktuellen Geschwindigkeit des Pkw. Die Eingangsgrößen der Regelung sind die Eigengeschwindigkeit des Fahrzeugs, sein Abstand und seine Relativgeschwindigkeit zum vorausfahrenden Fahrzeug. Eine der unscharfen Regeln lautet:

WENN Fahrzeugabstand = mittel
UND Relativgeschwindigkeit = stark negativ
UND Eigengeschwindigkeit = hoch
DANN Sollbeschleunigung = negativ

Den variablen Größen Fahrzeugabstand, Relativgeschwindigkeit, Eigengeschwindigkeit und Sollbeschleunigung werden jedoch keine numerischen Werte zugewiesen, sondern Ausdrücke der natürlichen Sprache. Im vorliegenden Beispiel werden diese Größen als *linguistische Variable* modelliert und die Werte, die sie annehmen können, als *linguistische Werte* [179, 180].

Definition 2.13: *Unscharfe Variable, unscharfe Beschränkung, Verträglichkeitsgrad.*

Eine *unscharfe Variable* („fuzzy variable“) wird durch ein Tripel $(X, U, R(X, u))$ charakterisiert. X ist der Name der unscharfen Variable, U ist die Grundmenge, u ist ein generischer Name für alle Elemente von U und $R(X, u)$, kurz $R(X)$, ist eine unscharfe Menge über U , die eine *unscharfe Beschränkung* („fuzzy restriction“) darstellt, die den Werten u durch die Variable X auferlegt ist [21]. Ist x ein generischer Name für alle Werte von X , dann bezeichnet

$$x = u : R(X, u) \tag{2.14}$$

eine Zuweisung des Wertes u an x unter der Beschränkung $R(X)$. Der Grad, zu welchem diese Gleichung erfüllt ist, wird als *Verträglichkeitsgrad* („compatibility degree“) $c(u)$ bezeichnet und es gilt

$$c(u) := \mu_{R(X)}(u), u \in U.$$

[180]

□

Beispiel 2.1: Der Begriff „mittel“ in obiger Regel bzw. „mittelgroßer Fahrzeugabstand“ ist eine unscharfe Variable $(X, U, R(X, u))$. Ihr Name X ist „mittel“. Die Grundmenge U ist ein

Entfernungsintervall, das unter praktischen Gesichtspunkten keine negativen Entfernungen enthalten muss und keine Werte, die größer sind als die Reichweite des Radargeräts u_{\max} im Fahrzeug, das heißt $U = [0, u_{\max}]$. Und R ist eine unscharfe Menge aus $IF(U)$ mit einer dem Zusammenhang entsprechenden Definition, z. B. die in Abb. 2.1 dargestellte unscharfe Menge.

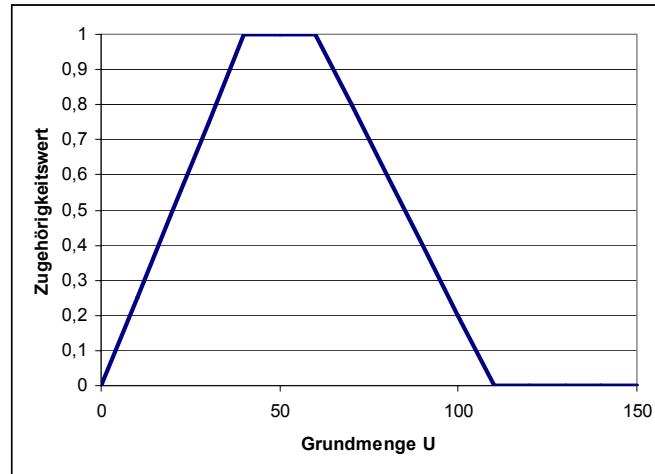


Abbildung 2.1: Beispiel der unscharfen Menge $R \in IF(U)$ zur Definition der unscharfen Variablen „mittel“ bzw. „mittlerer Fahrzeugabstand“.

Definition 2.14: *Linguistische Variable, Term-Menge, linguistischer Wert, Basisvariable, syntaktische Regel, semantische Regel.*

Eine *linguistische Variable* („linguistic variable“) wird durch ein Quintupel $(X, T(X), U, G, M)$ charakterisiert. Dabei ist X der Name der linguistischen Variablen. $T(X)$ bezeichnet die Menge der Terme von X - die *Term-Menge*. Die Terme sind die Namen der *linguistischen Werte* von X , wobei jeder Wert eine unscharfe Variable über der Grundmenge U mit der *Basisvariablen* u darstellt. G ist eine Menge *syntaktischer Regeln* zur Erzeugung der Term-Menge und M eine Menge *semantischer Regeln*, mit denen jedem Term Y eine unscharfe Menge $M(Y)$ über U als *Bedeutung* zugewiesen wird.

Die Bedeutung $M(Y)$ des Terms Y ist gegeben durch eine Restriktion $R(Y)$ vermöge der unscharfen Variablen mit dem Namen Y . Deshalb ist

$$M(Y) = R(Y),$$

wobei $R(Y)$ – und damit $M(Y)$ – als unscharfe Variable über U mit dem Namen Y betrachtet wird [180]. \square

Beispiel 2.2: „Fahrzeugabstand“ ist eine linguistische Variable $(X, T(X), U, G, M)$. Ihr Name X ist „Fahrzeugabstand“, ihre Termmenge $T(X)$ enthält z. B. „klein“, „mittel“, „groß“ und die Oder-Verknüpfungen von „klein“ und „mittel“ und „mittel“ und „groß“. Die Grund-

menge U sei dieselbe wie in Bsp. 2.1. Die syntaktischen Regeln in G besagen, dass nur „klein“ und „mittel“ bzw. „mittel“ und „groß“ durch „oder“ verknüpft werden dürfen. Die Menge M enthält für die linguistischen Variablen „klein“, „mittel“ und „groß“ die unscharfen Mengen R_{klein} , R_{mittel} und $R_{\text{groß}}$ aus $IF(U)$ (siehe Abb. 2.2) sowie die Festlegung, dass sich die Bedeutung von Oder-Verknüpfungen aus der unscharfen Vereinigung der den verknüpften Termen entsprechenden unscharfen Mengen gemäß Def. 2.11 und für die Minimum-t-Norm ergibt. Die unscharfen Mengen $R(\text{klein oder mittel})$ und $R(\text{mittel oder groß})$ sind in Abb. 2.3 dargestellt.

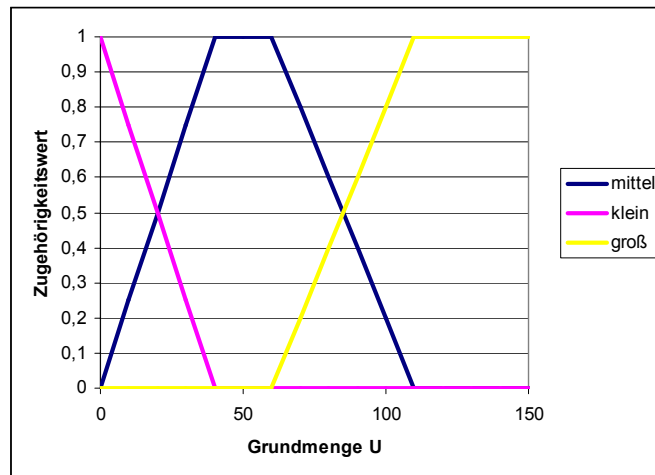


Abbildung 2.2: Die unscharfen Mengen $R(\text{klein})$, $R(\text{mittel})$ und $R(\text{groß})$ der linguistischen Werte „klein“, „mittel“ und „groß“ der linguistischen Variablen „Fahrzeugabstand“.

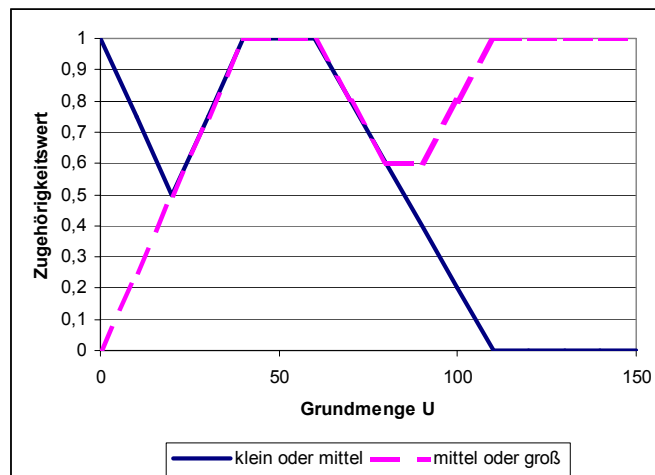


Abbildung 2.3: Die unscharfen Mengen $R(\text{klein oder mittel})$ und $R(\text{mittel oder groß})$ der linguistischen Werte „klein oder mittel“ und „mittel oder groß“ der linguistischen Variablen „Fahrzeugabstand“.

Anmerkungen zu Def. 2.14

1. Zadeh unterscheidet in der Termmenge zwischen *Grundtermen* und *zusammengesetzten Termen*. Im einfachsten Fall ist die Termmenge auf eine Menge von Grundtermen begrenzt.
2. Der Gebrauch linguistischer Variabler über einer Grundmenge U an Stelle numerischer Werte aus U kann als eine Form der Datenkomprimierung betrachtet werden, die auch als *Granulierung* („granulation“) bezeichnet wird. Gegenüber der Unterteilung einer Zahlenmenge in Intervalle – ebenfalls eine Datenkomprimierung – hat die Granulierung mit einander überlappenden unscharfen Mengen folgende Vorteile: [185]
 - Sie ist allgemeiner;
 - sie entspricht der Art und Weise, wie Menschen linguistische Werte interpretieren, nämlich eher als unscharfe Mengen denn als Intervalle;
 - der Übergang von einem linguistischen Wert zu einem benachbarten erfolgt nicht abrupt, sondern gleitend.

Beispiel 2.3: Der Gebrauch der linguistischen Variablen „Fahrzeugabstand“ in der unscharfen Regel auf Seite 28 mit den linguistischen Werten aus Bsp. 2.2 ist eine Granulierung gegenüber dem Gebrauch der in Metern gemessenen physikalischen Größe „Fahrzeugabstand“. Als physikalische Größe kann der in Metern gemessene Abstand unendlich viele Werte aus $U = [0, 150]$ annehmen. Als linguistische Variable kann er nur die fünf linguistischen Werte seiner Termmenge annehmen.

Große Bedeutung bei der Bildung von zusammengesetzten Termen aus Grundtermen haben linguistische Modifikatoren⁸⁾.

Definition 2.15: *Linguistischer Modifikator.*

Linguistische Modifikatoren sind sprachliche Ausdrücke, deren Funktion darin besteht, andere Ausdrücke abzustumpfen oder deren Unschärfegrad zu modifizieren. □

Typische Beispiele linguistischer Modifikatoren im Deutschen sind: sehr, besonders, relativ, eine Art, grob gesprochen, mehr oder weniger, ziemlich, praktisch, in mancher Hinsicht und dergleichen⁹⁾.

2.2 Zur Kritik an der unscharfen Logik

In der Geschichte der Missverständnisse über die unscharfe Logik, der Kritik an ihr und der Angriffe gegen sie nimmt die Wahrscheinlichkeitsrechnung einen besonderen Platz ein, worauf im Abschnitt 2.4 näher eingegangen wird. Die Geschichte der Auseinandersetzung mit der unscharfen Logik und ihre verschiedenen Seiten sind in [110] dargestellt. Den Kritikern, die

8. Linguistischer Modifikator ist eine andere Bezeichnung für „linguistische Hecken“ („linguistic hedges“), die in der Literatur ebenfalls gebräuchlich ist [193, 21].

9. Eine umfangreiche Aufzählung deutscher und englischer linguistischer Modifikatoren sowie eine Klassifikation und Modellierungsansätze für linguistische Modifikatoren enthält [21].

die Unschärfe selbst oder die Notwendigkeit unscharfer Verfahren in Frage stellen, hält Engels lange vor Begründung der unscharfen Logik entgegen:

"Für den Metaphysiker¹⁰⁾ sind die Dinge und ihre Gedankenabbilder, die Begriffe, vereinzelte, eins nach dem andern und ohne das andre zu betrachtende, feste, starre, ein für allemal gegebne Gegenstände der Untersuchung. Er denkt in lauter unvermittelten Gegensätzen: seine Rede ist ja, ja, nein, nein, was darüber ist, ist vom Übel. Für ihn existiert ein Ding entweder, oder es existiert nicht: ein Ding kann ebensowenig zugleich es selbst und ein andres sein. Positiv und negativ schließen einander absolut aus; Ursache und Wirkung stehn ebenso in starrem Gegensatz zueinander. Diese Denkweise erscheint uns auf den ersten Blick deswegen äußerst plausibel, weil sie diejenige des sogenannten gesunden Menschenverstandes ist. Allein der gesunde Menschenverstand, ein so respektabler Geselle er auch in dem hausbackenen Gebiet seiner vier Wände ist, erlebt ganz wunderbare Abenteuer, sobald er sich in die weite Welt der Forschung wagt; und die metaphysische Anschauungsweise, auf so weiten, je nach der Natur des Gegenstandes ausgedehnten Gebieten sie auch berechtigt und sogar notwendig ist, stößt doch jedesmal früher oder später auf eine Schranke, jenseits welcher sie einseitig, borniert, abstrakt wird und sich in unlösliche Widersprüche verirrt, weil sie über den einzelnen Dingen deren Zusammenhang, über ihrem Sein ihr Werden und Vergehen, über ihrer Ruhe ihre Bewegung vergißt, weil sie vor lauter Bäumen den Wald nicht sieht." [48]

Ein Beispiel für die Kritik der Anfangsjahre ist die äußerst emotionale und feindselige Stellungnahme von Kalman zu Zadehs Ideen im Jahr 1972:

„No doubt Professor Zadeh's enthusiasm for fuzziness has been reinforced by the prevailing political climate in the US – one of unprecedented permissiveness. 'Fuzzification' is a kind of scientific permissiveness; it tends to result in socially appealing slogans unaccompanied by the discipline of hard scientific work and patient observation. I must confess that I cannot conceive of 'fuzzification' as a viable alternative for the scientific method; I even believe that it is healthier to adhere to Hilbert's naive optimism: 'Wir wollen wissen: wir werden wissen.'“ [182]

Eine jüngere, vielbeachtete Auseinandersetzung wurde von Elkan in seinem Artikel „The Paradoxical Success of Fuzzy Logic“ geäußert [46, 47]. Als Hauptargument gegen die unscharfe Logik führt er an, dass sich diese auf die zweiwertige Logik reduziere, wenn die Ausdrücke

$$\overline{A \wedge \bar{B}} \text{ und } B \vee (\bar{A} \wedge \bar{B})$$

als logisch äquivalent (für den Implikationsoperator) angenommen werden. Hinter dieser Äquivalenz verbirgt sich der Satz vom ausgeschlossenen Dritten, der in der unscharfen Logik nicht vorausgesetzt wird (vgl. Def. 1.1), wodurch sich Elkans Kritik selbst entkräftet¹¹⁾. Wei-

10. Es handelt sich hier um eine Auslegung des Wortes *Metaphysik* im Engelsschen Sinne, das heißt als Gegensatz zur marxistischen Dialektik, nicht im Sinne der griechischen oder der klassischen deutschen Philosophie.

tere erhitzte Debatten finden auch 35 Jahre nach Begründung der unscharfen Logik z. B. in Fuzzy-Mailing-Listen (z. B. [58]) statt.

Sachlicher ist die Diskussion in [145]. Darin stellt Mammitzsch drei Kriterien zur Bewertung der Theorie unscharfer Mengen auf [107]:

1. Innermathematisches Kriterium

„Ist die Fuzzy Theorie als in sich geschlossene mathematische Theorie der Stochastik vergleichbar?“

2. Wissenschaftstheoretisches Kriterium

„Gibt es unterschiedliche Konzepte von Unsicherheit, die ihrem Wesen nach so verschieden sind, daß sie unterschiedliche mathematische Modelle erfordern?“

3. Pragmatisches Kriterium

„Gibt es sachwissenschaftliche Probleme, für die sich mittels der Fuzzy Theorie einfachere mathematische Modelle als mit Hilfe der Stochastik aufstellen lassen, und liefern die erzielten Ergebnisse eine einfachere sachwissenschaftliche Interpretation?“

So unterschiedlich die Antworten ausfallen, so günstig sind sie für die Theorie unscharfer Mengen. Mammitzsch verneint die erste Frage, gibt aber zu bedenken, dass die Theorie unscharfer Mengen verhältnismäßig jung sei, während die Stochastik erst nach fast 300 Jahren ihre endgültige Ausprägung gefunden hätte. Bei der zweiten Frage handle es sich eher um eine philosophische Frage, so dass keine mathematische Antwort darauf gefunden werden könne. Und die dritte Frage bejaht er mit Hinweis auf zahlreiche Anwendungen.

2.3 Unscharfe Logik und mehrwertige Logik

Die eigentliche Einführung der mehrwertigen Logik beginnt mit Artikeln von Lukasiewicz [104] und Post in den 20er Jahren [136]. Über die von ihm entwickelten mehrwertigen Logikkalküle sagt Lukasiewicz:

„Es war mir von vornherein klar, dass unter allen mehrwertigen Systemen nur zwei eine philosophische Bedeutung beanspruchen können: das dreiwertige und das unendlichwertige System. Denn werden die von '0' und '1' verschiedenen Werte als 'das Mögliche' gedeutet, so können aus guten Gründen nur zwei Fälle unterschieden werden: entweder nimmt man an, dass das Mögliche keine Gradunterschiede aufweist, und dann erhält man das dreiwertige System; oder man setzt das Gegenteil voraus, und dann ist es am natürlichsten ebenso wie in der Wahrscheinlichkeitsrechnung anzunehmen, dass unendlich viele Gradunterschiede des Möglichen bestehen, was zum unendlichwertigen Aussagenkalkül führt. Ich glaube, dass gerade dieses letztere System vor allen anderen den Vorzug verdient. Leider ist dieses System noch nicht genau untersucht; insbesondere ist auch das Verhältnis des unendlichwertigen Systems zur Wahrscheinlichkeitsrechnung noch nicht geklärt.“ [106]

-
11. Unter der Annahme der Gültigkeit des Satzes vom ausgeschlossenen Dritten reduziert sich auch eine mehrwertige Logik auf eine zweiwertige Logik.

Das Zitat legt den Bezug der mehrwertigen Logik zur unscharfen Logik nahe, die unendlich viele Wahrheitswerte (alle Elemente des Einheitsintervalls $[0, 1]$) kennt. Zugleich verdeutlicht es, dass der Ausgangspunkt von Lukasiewicz Überlegungen ein anderer ist als bei der unscharfen Logik. Zadehs Ausgangspunkt in seiner ersten Arbeit „Fuzzy Sets“ [177] ist die Tatsache, dass Klassen von Gegenständen in der realen Welt häufig keine genauen Zugehörigkeitskriterien besitzen (vgl. Abschnitt 1.1). Der Ausgangspunkt für Lukasiewicz Ideen war dagegen das Problem der *Möglichkeit* von Aussagen und das Problem des *Determinismus*, die miteinander zusammenhängen:

„Der 'Zweiwertigkeitssatz' ist die tiefste, jedoch schon im Altertum heftig umstrittene Grundlage unserer gesamten Logik. ... Der Streit um den Zweiwertigkeitssatz hat einen metaphysischen Hintergrund: die Anhänger dieses Satzes sind entschiedene Deterministen, während die Widersacher des Satzes zur indeterministischen Weltanschauung hinneigen.“ [106]¹²⁾

Wie im Abschnitt 2.1 bereits ausgeführt, hat unscharfe Logik zahlreiche, recht unterschiedliche Aspekte. Darunter hat die *unscharfe Logik im engen Sinn* den deutlichsten Bezug zur mehrwertigen Logik. In [13] wird dazu näher ausgeführt:

„It is important to observe that fuzzy logic ... is a generic term which refers not to a unique logical system but to a collection of local logics¹³⁾ in which the truth-values are fuzzy subsets of the truth value set of an underlying multivalued logic. For example, if the underlying logic ... is Lukasiewicz's L_{\aleph_1} logic, then the truth-values of a fuzzy logic whose base logic is L_{\aleph_1} would be fuzzy subsets of the unit interval.“

Die unscharfe Logik im engen Sinn betrachtet also Wahrheit als *linguistische Variable* im Sinne der Def. 2.14 und spricht daher auch von *linguistischer Wahrheitsvariable* („linguistic truth variable“) und *linguistischen Wahrheitswerten* („linguistic truth values“), worunter sie Terme, wie z. B. sehr wahr, ziemlich wahr, mehr oder weniger wahr, im wesentlichen wahr, falsch, völlig falsch usw. versteht [180].

Gemeinsamkeiten von unscharfer und mehrwertiger Logik bestehen bei den logischen Funktoren für Aussagen. So sind in der unscharfen Logik einige von Lukasiewicz eingeführte Funktoren gebräuchlich¹⁴⁾, z. B.: $u \mathbf{t} v := \max\{0, u + v - 1\}$, $u \mathbf{s} v := \min\{1, u + v\}$ und $u \mathbf{\phi} v := \min\{1, 1 - u + v\}$.

Wird der Maßstab des Anwendungsbezugs angelegt, dann hat die unscharfe Logik augenscheinlich mehr Erfolg als die mehrwertige Logik. Ohne ein abschließendes Urteil fällen zu wollen, vermutet der Verfasser, dass die Ursache dafür in den unterschiedlichen Ausgangspunkten von Lukasiewicz und Zadeh liegt. Während der Mathematiker Lukasiewicz über die Betrachtung philosophischer Probleme zur dreiwertigen Logik kommt, geht der Elektrotechni-

12. Siehe auch die Eröffnungsrede von Lukasiewicz an der Warschauer Universität zum akademischen Jahr 1922/23 [105].

13. Die unscharfe Logik wird hier als „lokal“ bezeichnet, weil die Bedeutung ihrer Aussagen, Operatoren und Wahrheitswerte nur „lokal“, das heißt in einem festgelegten (Untersuchungs-) Bereich gültig sind.

14. Umgekehrt, und allgemeiner, lässt sich feststellen, dass alle mehrwertigen Lukasiewicz-Logiken t-Normbasiert sind [66].

ker Zadeh vom anschaulichen Problem der Beschreibung von Unschärfe in der realen Welt aus. Und während das Verständnis von Lukasiewiczs Logikkalkülen überwiegend Mathematikern vorbehalten bleibt, entwickelt Zadeh mit der *unscharfen Menge* (vgl. Def. 2.1) einen Formalismus, der am leicht verständlichen Mengenbegriff ansetzt und mit Hilfe des *Erweiterungsprinzips* (vgl. Def. 2.12) in ganz unterschiedlichen Bereichen angewandt werden kann.

2.4 Unscharfe Logik und Wahrscheinlichkeitsrechnung

Unter den verschiedenen Wissenschaften gehören die Mathematiker und insbesondere die Stochastiker zu den eifrigsten Kritikern der unscharfen Theorie¹⁵⁾. Über das Thema Wahrscheinlichkeit und Unschärfe sind mehrere hundert Artikel erschienen [19]. Die Fortführung dieser Diskussion ist nicht der Gegenstand dieser Arbeit. Weil hier aber Elemente der Theorie der unscharfen Menge angewendet werden, ist es notwendig, die Auseinandersetzung zwischen Wahrscheinlichkeitsrechnung und unscharfer Logik kurz zu beleuchten und den Standpunkt des Verfassers darzulegen.

Der Kern der Kritik von Vertretern der Wahrscheinlichkeitsrechnung besteht in der Behauptung, dass die Wahrscheinlichkeitsrechnung ausreichend sei, um Unschärfe zu beschreiben¹⁶⁾. Dem steht der Standpunkt Zadehs gegenüber, dass Wahrscheinlichkeitsrechnung und unscharfe Logik einander ergänzen [184]. Zum besseren Verständnis sei stellvertretend für die Vertreter des Alleinvertretungsanspruchs der Wahrscheinlichkeitsrechnung auf [7] und [170] verwiesen. Neben pauschalen Behauptungen, dass unscharfe Konzepte, wie das des Zugehörigkeitswertes eines Elements zu einer Menge keine fundierte Grundlage hätten bzw. nur „verkleidete“ Wahrscheinlichkeitswerte („probabilities in disguise“) seien, wird Aussagen von Vertretern der unscharfen Logik über die Überlegenheit ihrer Theorie unterstellt, sie beruhten auf Un- und Missverständnissen gegenüber stochastischen Methoden, und schließlich wird beispielhaft gezeigt, wie Mamdani-Regler unter bestimmten Voraussetzungen als stochastische Regler interpretiert werden können und darauf verwiesen, dass auch stochastische Regler universelle Approximatoren seien. Abgesehen von einem Verweis auf Engels Kritik am metaphysischen Standpunkt (vgl. Zitat in Abschnitt 2.1) lohnt es sich nicht, im Einzelnen auf diese Einwendung einzugehen. Stattdessen wird nachfolgend der Standpunkt zum Verhältnis von Wahrscheinlichkeitsrechnung und unscharfer Logik aus der Sicht der unscharfen Logik dargestellt. Zuvor sei aber auf eine historische Parallele hingewiesen.

Beim Studium der Literatur aus den 20er und 30er Jahren über mehrwertige Logik ist festzustellen, dass das Verhältnis der mehrwertigen Logik zur Wahrscheinlichkeitsrechnung damals genauso thematisiert bzw. problematisiert wurde wie das Verhältnis der unscharfen Logik zur Wahrscheinlichkeitsrechnung in den letzten 30 Jahren¹⁷⁾. In [192] wird dieses Verhältnis expli-

15. Es gibt auch einige Ausnahmen. Zum Beispiel vertritt Viertl die Auffassung, dass jede konkrete Messung eine unscharfe Zahl ist, die noch mit einem (stochastischen) Fehler versehen sein kann. Daraus entwickelt er Ansätze für die statistische Analyse unscharfer Daten [154, 156]. (Anmerkung: Unscharfe Zahlen sind besondere unscharfe Mengen.)

16. Für die Grundlage dieser Ansicht gibt es zwei Möglichkeiten. Erstens, die Anschauung, dass nicht-statistische Ungenauigkeit nicht existiert (vgl. dazu Abschnitt 2.2). Und zweitens, die Meinung, dass nicht-statistische Ungenauigkeit zwar existiert, ein stochastisches Modell aber die beste Wahl für ihre Modellierung ist [19].

17. Zu dieser Analogie sind dem Verfasser keine Hinweise in der Literatur über unscharfe Logik bekannt.

zit thematisiert. Der Ausgangspunkt dabei ist die Frage: „*Welche Rolle spielen aber hier diese Zahlen?*“ (Gemeint sind die Wahrheitswerte zwischen Null und Eins.) Und [136] zitiert Z. Jordan¹⁸⁾:

„The difficulty with the n -valued systems does not consist so much in technical problems, considerable as these are, as in finding an interpretation of the n „truth-values“ involved in the system. Without an interpretation assigning a definite logical meaning to the n „truth-values“ any given n -valued calculus remains an abstract structure.“ (a. a. O.)

Zawirski stellt weiterhin fest:

„Die grösste Schwierigkeit auf dem Wege der Versöhnung der neuen Logik mit dem W.-Kalkül¹⁹⁾ bilden jedoch die Formeln für die Summe und das Produkt der Aussagen, welche mit den diesbezüglichen Formeln der W.-Rechnung nicht übereinstimmen.“ (a. a. O.)

Das Zitat zeigt, dass damals wie heute der Wunsch bestand, die Semantik der Wahrheitswerte bzw. Zugehörigkeitswerte dadurch zu lösen, dass die neue Theorie auf Bekanntes zurückgeführt wird. Die Methoden, die bei dieser „Versöhnung“ angewandt werden, sind ebenfalls verwandt. Bei Lukasiewicz nehmen die Formeln

$$p \vee \neg p \text{ und } \neg(p \wedge \neg p) \quad (2.15)$$

für $p = 1/2$ den Wert $1/2$ an. Zawirski stellt fest, dass das mit einer Interpretation der Wahrheitswerte als Wahrscheinlichkeiten nicht vereinbar ist und unterstellt Lukasiewicz bei der Definition der Disjunktion

$$p \vee q := (p \rightarrow q) \rightarrow q$$

„eine gewisse Willkür“ (a. a. O.). Dann definiert er für die Disjunktion

$$p \vee q := \neg p \rightarrow q ,$$

wodurch die Formeln in (2.15) auch für $p = 1/2$ den Wert 1 annehmen. Dass dabei der Satz vom ausgeschlossenen Dritten trotz der Dreiwertigkeit wieder Gültigkeit erhält – was der Intention von Lukasiewicz widerspricht (vgl. [104]) – stört Zawirski nicht. Dieses Vorgehen zeigt große Verwandtschaft mit der Argumentation in Elkans Kritik an der unscharfen Logik (vgl. Abschnitt 2.2).

Zur Abgrenzung der unscharfen Logik gegen die Wahrscheinlichkeitsrechnung bedienen sich die Vertreter der unscharfen Logik meist dreier Methoden:

1. Sie führen Beispiele für Unschärfe an, die mit der Wahrscheinlichkeitsrechnung nicht beschreibbar sind.
2. Sie vergleichen Unschärfe und Wahrscheinlichkeit direkt anhand von Beispielen.

18. [82]

19. W.-Kalkül = Wahrscheinlichkeitskalkül

3. Sie widerlegen stochastische Interpretationen von Zugehörigkeitswerten.

Bezüglich des letzten Punktes gibt es bei den Vertretern der unscharfen Logik unterschiedliche Standpunkte. Während der eine eine stochastische Interpretation von Zugehörigkeitswerten grundsätzlich ablehnt, weil solche Interpretationen nicht möglich seien, lässt der andere solche Interpretationen für bestimmte Modelle und Umstände zu.

Zadeh vertritt die Meinung, dass

- Wahrscheinlichkeitsrechnung und unscharfe Logik verschieden sind,
- Wahrscheinlichkeitsrechnung nicht ausreicht, um mit Unsicherheit („uncertainty“) und Ungenauigkeit („imprecision“) zurechtzukommen,
- Wahrscheinlichkeitsrechnung und unscharfe Logik sich gegenseitig ergänzende und nicht sich gegenseitig ausschließende Disziplinen sind [184].

Zur Begründung seines Standpunkts greift Zadeh auf die oben genannten drei Methoden zurück. Bezüglich Punkt 1 führt er unter anderem folgende Beispiele an [184]:

- a) *Unscharfe Ereignisse* können mit der Wahrscheinlichkeitsrechnung nicht beschrieben werden. Einfache Beispiele von Aussagen, die unscharfe Ereignisse einschließen, sind:
 - In naher Zukunft wird es *ein starkes Erdbeben* geben.
 - Morgen wird *ein warmer Tag*.
- b) *Unscharfe Quantoren* („fuzzy quantifiers“) können mit der Wahrscheinlichkeitsrechnung nicht beschrieben werden, z. B. *viel, die meisten, einige, wenige*.
- c) Berechnungen mit *unscharfen Wahrscheinlichkeiten* sind mit der Wahrscheinlichkeitsrechnung nicht möglich. Unscharfe Wahrscheinlichkeiten sind z. B.: *wahrscheinlich, unwahrscheinlich, nicht sehr wahrscheinlich*.
- d) Die Ausdruckskraft der Wahrscheinlichkeitsrechnung reicht für eine Bedeutungsdarstellende Sprache („meaning-representation language“) nicht aus. Beispiel: Was ist die Bedeutung von: „*Es ist nicht wahrscheinlich, dass es in naher Zukunft einen scharfen Anstieg des Ölpreises geben wird.*“

Zu den beispielhaften Vergleichen von Wahrscheinlichkeitsrechnung und unscharfer Logik (Punkt 2) gehört

Beispiel 2.4: Gegeben sei die Aussage: „Sie sind aus Claras Auto (VW Käfer) ausgestiegen.“ [181]. Es stellt sich die Frage: „Wieviele Mitfahrer (Clara, die Fahrerin eingeschlossen) sind aus Claras Auto ausgestiegen?“ In diesem Beispiel interpretiert Zadeh den Zugehörigkeitswert zu einer unscharfen Menge als Grad der Möglichkeit. Bezeichne n eine Anzahl von Mitfahrern und $\mu(n)$ und $P(n)$ die Möglichkeit bzw. Wahrscheinlichkeit, mit welcher n Mitfahrer aus Claras Auto aussteigen. Für $\mu(n)$ und $P(n)$ sind die Werte in Tabelle 2.1 gegeben.

Der Grad der Möglichkeit, mit dem 5 Mitfahrer aus Claras Auto aussteigen, das heißt der Grad der Einfachheit, mit dem sich 5 Mitfahrer in einen VW Käfer quetschen können, beträgt somit 0,7. Andererseits ist die Wahrscheinlichkeit, mit der Clara 5 Menschen in ihrem Auto mitnimmt, gleich null. Für $n = 1$ sind sowohl die Möglichkeit als auch die Wahrscheinlichkeit

	n	1	2	3	4	5	6	7
Möglichkeit	$\mu(n)$	0	1	1	1	0,7	0,2	0
Wahrscheinlichkeit	$P(n)$	0	0,6	0,3	0,1	0	0	0

Tabelle 2.1: Beispiel für den Unterschied zwischen Möglichkeit und Wahrscheinlichkeit.

gleich null, da in der Aussage vorausgesetzt wird, dass Clara nicht alleine gefahren ist. Das Beispiel zeigt, dass Möglichkeit keine zweiwertige Eigenschaft ist und dass Möglichkeitsgrade nicht dasselbe wie Wahrscheinlichkeiten sind. Und schließlich wird deutlich, dass möglichkeitsbasierte Information elementarer und weniger kontextabhängig als wahrscheinlichkeitsbasierte Information ist.

Die stochastische Interpretation von Zugehörigkeitswerten wird hier nur kurz angerissen werden, weil ihre umfassende Darstellung aus Platzgründen nicht möglich ist. Als Beispiel für die Widerlegung einer solchen Interpretation (Punkt 3) wird das Modell von Hisdal [75] und Zadehs Kritik daran [184] kurz vorgestellt. Hisdals Modell gehört zu den so genannten Abstimmungsmodellen („voting models“).

Beispiel 2.5: Gegeben sei die unscharfe Menge A durch ihre Zugehörigkeitsfunktion μ_A über der Grundmenge X , und V sei eine Menge von Wählern²⁰⁾ $\{V_1, \dots, V_n\}$, von denen ein jeder abstimmt, ob ein gegebener Wert $u \in X$ als A oder nicht A klassifiziert werden soll. Dann ist $P(A | u)$ die bedingte Wahrscheinlichkeit, dass ein zufällig aus der Menge V herausgegriffener Wähler den Wert $u \in X$ als A klassifiziert. In diesem Sinne bedeutet

$$\mu_{\text{jung}}(25) = 0,8 ,$$

dass der Anteil der zufällig herausgegriffenen Wähler, die 25 Jahre als „jung“ klassifizieren würden, gleich 0,8 ist.

Gegen dieses Modell wendet Zadeh ein,

- dass es unnatürlich ist, wenn ein Wähler bezüglich eines unscharfen Begriffes A nur mit A oder nicht A antworten kann, und
- dass Modelle dieser Art nicht greifen, wenn subjektive Bewertungen stattfinden, wie z. B.: „Zu welchem Grad ist Clara nach Ihrer Ansicht groß?“

Weitere Argumente gegen solche Modelle finden sich in [68].

Zadeh akzeptiert jedoch eine Interpretation von Zugehörigkeitswerten mittels Zufallsmengen („random sets“) [184], weist jedoch darauf hin, dass trotz der Verbindung von Zufallsmengen und unscharfer Logik Beides unterschiedliche Konzepte sind.

Außer den hier erwähnten Interpretationen unscharfer Mengen mit Hilfe von Zufallsmengen oder Abstimmungsmodellen gibt es weitere Verfahren, denen andere Betrachtungsweisen zu

20. Im Folgenden wird alternativ auch die Bezeichnung „Abstimmender“ gebraucht.

Grunde liegen. So sieht Sustal [149] die Schärfung einer unscharfen Menge (in einem Regelbasierten unscharfen System) als Filterung: Zu einer gegebenen unscharfen Menge wird zufällig ein α -Schnitt (vgl. Def. 2.5) bestimmt, und aus diesem α -Schnitt – der eine scharfe Menge ist – wird zufällig ein Wert herausgegriffen. Dabei folgt die Wahl des α -Schnitts einer gegebenen Wahrscheinlichkeitsdichte, die Sustal als Verträglichkeitsgrad („compatibility degree“) bezeichnet. Und die Wahl des Wertes aus dem α -Schnitt folgt einer Gleichverteilung. Durch Umkehrung dieses Prozesses kann unter bestimmten Bedingungen aus einer gegebenen Wahrscheinlichkeitsdichte eine unscharfe Menge berechnet werden (Einzelheiten in [149]).

2.5 „Soft computing“ und „Computing with words“

Das bereits in Abschnitt 1.1 aufgeworfene Problem:

„Insofern sich die Sätze der Mathematik auf die Wirklichkeit beziehen sind sie nicht sicher, und insofern sie sicher sind, beziehen sie sich nicht auf die Wirklichkeit.“ [45]

beantwortet Einstein bezüglich der Geometrie (= Mathematik) einerseits und physikalischer Lagemessungen (= Wirklichkeit) andererseits so:

„... es fällt nicht schwer, den physikalischen Zustand eines Meßkörpers so genau festzulegen, daß sein Verhalten bezüglich der relativen Lagerung zu anderen Meßkörpern hinreichend eindeutig wird, so daß man ihn für den „starren“ Körper (der Geometrie, Anm. d. Verf.) substituieren darf.“ (Unterstreichung durch d. Verf.)

Was „hinreichend“ ist, hängt aber vom Zweck der Messung ab. Der Kern der Aussage ist, dass ein vereinfachtes Modell ausreicht, wenn es den Anforderungen genügt, die ihm gestellt werden. Bellman formuliert es so:

„We must balance the needs for exactness and simplicity, and reduce complexity without oversimplification in order to match the level of detail at each step with the problem we face.“ [12]

Den Gedanken, Genauigkeit zugunsten von Machbarkeit zu opfern, greift „soft computing“ auf und formuliert ihn auf höherer Stufe neu. Dabei geht es von den Grundsätzen aus,

- a) dass die reale Welt von Ungenauigkeit und Unsicherheit durchdrungen ist,
- b) Genauigkeit und Sicherheit aber „Kosten“ verursachen, z. B. hoher Zeitaufwand für die Modellbildung, großer Berechnungsaufwand für das Modell u. dgl.

Während traditionelle Berechnungsverfahren („hard computing“) Genauigkeit und Sicherheit verlangen und dafür hohe Kosten in Kauf nehmen, ist es die Richtschnur von „soft computing“, vorhandene Toleranzen für Ungenauigkeit, Unsicherheit und Teilwahrheiten auszunutzen, wenn dadurch die Bearbeitbarkeit und Robustheit erhöht und geringe Lösungskosten erreicht werden. Zadeh meint, das sei letztlich eine Nachahmung des menschlichen Gehirns, das die Fähigkeit für approximatives Schließen sehr effektiv einsetze [183, 185].

„Soft computing“ umfasst im Wesentlichen vier Bereiche [187]:

-
- Unschärfe Logik: approximatives Schließen, Informationsgranulierung, „computing with words“ (siehe unten);
 - neuronales Rechnen („neurocomputing“): Lernen, Adaption, Klassifikation, Systemmodellierung und -identifikation;
 - genetisches Rechnen („genetic computing“): Synthese, Feinabstimmung und Optimierung durch systematische zufallsbestimmte Suche und Evolution;
 - wahrscheinlichkeitsbasiertes Rechnen („probabilistic computing“): Handhabung von Unsicherheit, Vorhersage, chaotische Systeme, „belief networks“.

„Computing with words“ ist ähnlich motiviert wie „soft computing“, hat aber eine spezifischere Form. Es ist eine Methode, deren Berechnungen und Schlüsse mit Worten anstatt mit Zahlen operieren, und die Zadeh dann als notwendig erachtet,

- wenn die verfügbare Information zu ungenau ist, um den Gebrauch von Zahlenwerten zu rechtfertigen und
- wenn eine Toleranz für Ungenauigkeit ausgenutzt werden kann, um Bearbeitbarkeit, Robustheit und Lösungen mit geringen Kosten und eine bessere Übereinstimmung mit der Wirklichkeit zu erreichen.

Der Ausgangspunkt von „computing with words“ ist das Konzept des Körnchens („granule“), worunter Zadeh eine unscharfe Menge von Punkten versteht, die die Gestalt eines Klumpens hat, dessen Elemente durch Ähnlichkeit zusammengehalten werden. Ein Wort w ist der Name eines Körnchens g und umgekehrt ist g die Bedeutung von w .²¹⁾ Eine grundlegende Annahme von „computing with words“ ist, dass Information durch die Beschränkung der Werte von Variablen vermittelt wird. Ein Beispiel dafür ist (2.14): eine linguistische Variable ist eine durch eine unscharfe Menge beschränkte Basisvariable (= Variable einer scharfen Menge). Dieses Konzept verallgemeinert Zadeh auf andere Arten von Beschränkung, z. B. stochastische Beschränkungen, Beschränkungen durch raue Mengen („rough sets“) usw. [186, 191].

21. vgl. Def. 2.14: Der Gebrauch linguistischer Variabler kann als Granulierung betrachtet werden.

3 Modellbasierte Kundengewinnung

In diesem Kapitel wird für den Prozess der Kundengewinnung ein mathematisches Modell entwickelt. Mit seiner Hilfe kann der erwartete Verlauf einer Kundenakquise auf dem Computer simuliert werden und, was hier der Hauptzweck ist, das Modell liefert die Voraussetzung für die Lösung verschiedener Akquise-bezogener Optimierungsprobleme. Wichtige Grundlagen für die Abschnitte 3.1 – 3.3 sind die Arbeiten [32, 33, 34] und für den Abschnitt 3.4 [172].

Eines der Ziele im Marketing ist die Erzielung des größtmöglichen Gewinns durch Anwendung geeigneter Mittel zur Kundengewinnung, z. B. durch Massenbriefsendungen, so genannte „Mailings“, Briefe, Telefonate, Angebote und Produktvorführungen. In der Vergangenheit gehörte diese Aufgabe in die Hand von Männern und Frauen, die „eine Nase“, also ein „Gespür“ für das Verkaufen von Waren und Dienstleistungen hatten. Ein anschauliches, wenn auch fiktives Beispiel ist der Handlungsreisende Willy Loman aus Arthur Millers Bühnenstück „Tod eines Handlungsreisenden“ [113]. Das Stück zeigt auf tragische Weise, wie eng das Geschäftsergebnis von der richtigen Kundenansprache abhängt (auch wenn das nicht der Hauptgegenstand des Stückes ist). Willy Loman betont den subjektiven Faktor, indem er feststellt:

„Because the man who makes an appearance in the business world, the man who creates personal interest, is the man who gets ahead. ...

It's not what you say, it's how you say it – because personality always wins the day.“ [113]

Da heutzutage Mitarbeiter zunehmend als „Kostenfaktor“ gesehen werden, im verschärften Wettbewerb die Gewinnspannen einzelner Waren oder Dienstleistungen mitunter sehr klein sind und deshalb der Zwang besteht, möglichst viele (potenzielle) Käufer anzusprechen, das heißt einen möglichst großen Markt zu bearbeiten, ist die Kundengewinnung auf die Nutzung von Mitteln der Datenverarbeitung angewiesen¹⁾. Dazu gehört z. B. die Kundensegmentierung, die die großen Datenmengen nutzt, die üblicherweise in Datenbanken gespeichert sind, um Eigenschaften zu finden, die Kundengruppen charakterisieren und so die effektivsten Marketingmaßnahmen zu ergreifen [89, 92, 97, 98, 101, 151, 157, 153]. Der Schwerpunkt dieses Kapitels liegt auf einem anderen Gebiet, der mathematischen Modellierung der Kundengewinnung und, darauf aufbauend, der Optimierung von Marketing-Aktionen. Auf die vorhandenen Bezüge zur Kundensegmentierung wird an entsprechender Stelle jeweils hingewiesen. In den Abschnitten 5.2 und 5.3 von Kapitel 5 zeigt sich, dass die Modellierung der Kundengewinnung und die Kundensegmentierung eng zusammenhängen.

Die Beschäftigung mit dem Thema wurde durch die Zusammenarbeit des Lehrstuhls für numerische und angewandte Mathematik an der BTU Cottbus mit der Firma „Format“ in Frankfurt/Oder angeregt. Letztere entwickelte und vertreibt das kommerzielle Software-Werkzeug

1. [143] stellt fest, dass bedingt durch erhebliche Verlustzeiten für Vorbereitung und Verwaltung, für Reisen und Warten ein Verkäufer bestenfalls seine halbe Arbeitszeit für kommunikative Tätigkeiten aufwenden kann. „So kommt es, daß der Stundensatz eines Verkäufers bei 100 bis 150 DM liegt und daß ein Besuch eines Verkäufers etwa 300 DM und mehr kostet, wenn man sämtliche Aufwendungen auf die produktive Zeit umgelegt“ [a. a. O., S. 232].

AkquiSys, das die Planung, Organisation und Analyse der Kundengewinnung unterstützt, aber keine Optimierung der Marketing-Strategie erlaubt [30, 146, 57].

In diesem Kapitel wird ein mathematisches Modell für die Kundengewinnung entwickelt. Im Abschnitt 3.2 wird das mathematische Modell so genannter „Akquisepläne“ entwickelt, wie sie vom Software-Werkzeug *AkquiSys* verwendet werden. Das Modell ist die Grundlage für eine Simulation. Sie wird im Abschnitt 3.3 beschrieben und ermöglicht die Auswertung des zeitlichen Verlaufs von Größen der Kundengewinnung. Aufbauend auf dem Modell wird im Abschnitt 3.4 das Optimierungsproblem der „Maximalen Beschickung von Akquiseplänen“ formuliert. Dabei handelt es sich um ein unscharfes stochastisches Optimierungsproblem. Seine Lösung wird im folgenden Kapitel 4 behandelt und mit einem Fallbeispiel erläutert. Zu den Schwächen von Akquiseplänen gehört, dass sie keine Möglichkeit bieten, Kundeneigenschaften unmittelbar zu berücksichtigen. Eine Erweiterung, die dies ermöglicht, wird im Kapitel 5 entwickelt. Dabei wird die Kundengewinnung als mehrstufiger stochastischer Entscheidungsprozess mit unscharfem Ziel und unscharfen Nebenbedingungen modelliert, dessen Zustände durch Kundeneigenschaften beschrieben werden. In diesem Modell ist das Ziel (Optimierung der Marketingstrategie) die Berechnung einer optimalen Politik. Das Optimierungsproblem wird im Abschnitt 5.4 beschrieben und im Kapitel 6 gelöst und anhand eines Fallbeispiels erläutert.

3.1 Kundengewinnung mit Akquiseplänen

Kundengewinnung wird hier als Prozess verstanden, das heißt als Hintereinanderausführung von Marketing-Aktionen. Gewöhnlich beginnt der Prozess mit der Festlegung einer Zielgruppe, also potenzieller Kunden, die mit Marketing-Aktionen bearbeitet werden soll. Der Prozess endet entweder mit einem Verkauf oder Vertragsabschluss zwischen Akquisiteur²⁾ und Kunde. Unter Marketing-Aktionen werden alle Maßnahmen verstanden, die

- Informationen über Produkte – Güter oder Dienstleistungen – an den Kunden weitergeben,
- den Kunden direkt oder indirekt auffordern, auf für den Akquisiteur vorteilhafte Weise zu reagieren, z. B. Anforderung weiterer Produktinformationen, Anforderung eines Angebots, Kauf eines Produkts oder Bestellung einer Dienstleistung.

Beispiele für solche Aktionen sind Massenbriefsendungen, Briefe, Telefonate, Produktpräsentationen, Anzeigen usw. Da bekanntlich die Wirkung solcher Aktionen von Form und Inhalt abhängt, unterscheiden sich Aktionen, wenn sie in Form oder Inhalt verschieden sind, z. B. ein Brief mit allgemeinen Produktinformationen von einem Brief mit Detailinformationen.

Im Folgenden werden einmalige Akquisitionen betrachtet, also keine Dauer-Akquisitionen, bei denen über einen langen Zeitraum hinweg gleiche oder ähnliche Akquiseaktionen ausgeführt werden, wie z. B. beim Katalogversand des Versandhandels [62, 130, 131].

2. Im Folgenden wird der Begriff „Akquisiteur“ als Sammelbezeichnung für alle Kundengewinnung betreibenden Händler („marketer“) gebraucht.

Vereinfachende Bezeichnungsweise

Trotz ihres Anwendungsbezuges betrachtet die vorliegende Arbeit keine spezielle Branche der Wirtschaft. Branchenspezifische Ausdrücke sollen daher soweit wie möglich vermieden werden. Die mathematische Modellierung und Analyse der Prozesse stehen im Vordergrund. Deshalb und zur Vermeidung sprachlicher Ungelenkigkeiten werden folgende vereinfachende Ausdrücke verwendet:

- statt *Akquisition* einfach *Akquise*,
- statt *potenzieller Kunde* (also der beworbene Kunde im Gegensatz zum geworbenen Kunden) einfach *Kunde*,
- statt *Marketing-Aktion* (Marketing geht über das hier Betrachtete weit hinaus) einfach *Akquiseaktion* oder *Aktion*,
- statt *Marketing-Kampagne* einfach *Kundengewinnung* oder *Akquise*,
- statt *Kundenreaktion* einfach *Reaktion*,
- statt *Durchführung einer Akquise* oder *Akquiseprozess* einfach *Akquise*,
- und mit *Akquisiteur* werden gleichermaßen
 - der Entwickler eines Akquiseplans,
 - der Bediener der Software zum Entwurf und Abarbeiten eines Akquiseplans,
 - und der Verkäufer, Vertriebsmitarbeiter o. ä., der den Akquiseplan nutzt, bezeichnet.

3.1.1 Akquisepläne des Software-Werkzeugs AkquiSys

Das Software-Werkzeug *AkquiSys* ermöglicht zur Unterstützung der Kundengewinnung die Beschreibung so genannter Akquisepläne. Das *AkquiSys*-Handbuch führt dazu aus [57]:

„In der Akquiseplanung wird der Ablauf der Akquise festgelegt. Obwohl die Reaktion der potentiellen Kunden natürlich nicht vorhergesehen werden kann und somit ein Teil der Anfragen manuell erledigt werden muss, ist doch eine Vielzahl von Abläufen vorher bestimmbar. So kann z. B. nach der Rücksendung einer Postkarte die Versendung eines Prospektes ausgelöst werden bzw. bei Nichtbeantwortung eines Briefes nach einer Wartezeit automatisch ein zweiter Brief gedruckt werden oder die betreffende Adresse zu den Akten abgelegt werden. ...

Im Akquiseplan wird der Verlauf der Aktionen zur Kundengewinnung gezeigt. Hierzu müssen Sie erst einmal nach eigenen Wünschen ein Akquiseplan erstellen. Um sich ein Bild vom Aussehen eines Akquiseplan zu machen, wird Ihnen ein Beispiel mitgeliefert.“

Der Akquiseplan in Abb. 3.1 zeigt in den Quadraten und Rechtecken Marketing-Aktionen und in den Ovalen Kundenreaktionen, wobei zwischen direkten Reaktionen (gelbe Ovale) und indirekten Reaktionen (rote Ovale) unterschieden wird. Reaktionen bewirken einen Übergang von einer Aktion zur nächsten. Bei indirekten Reaktionen wird dieser Übergang dadurch ausgelöst, dass der Kunde nicht innerhalb eines vorgegebenen Zeitraums antwortet.

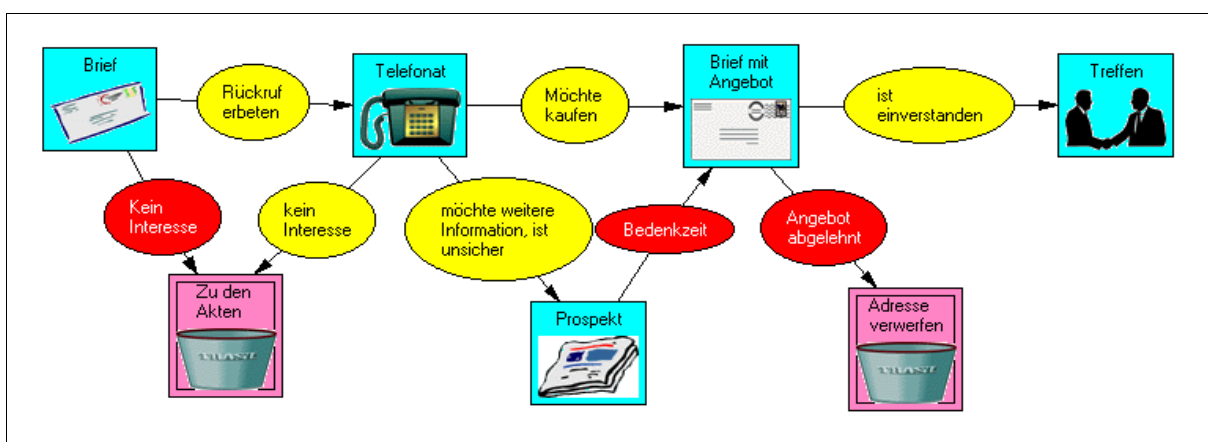


Abbildung 3.1: Beispiel eines mit *AkquiSys* erstellten und visualisierten Akquiseplans [57].

Kundensegmentierung im eigentlichen Sinn ist kein unmittelbarer Bestandteil einer mit einem Akquiseplan geplanten und durchgeführten Akquise. Wenn der Akquiseplan auf ein bestimmtes Kundensegment zugeschnitten ist, ist eine vorherige Kundensegmentierung aber nützlich, indem sie die zu bewerbende Kundengruppe auf diejenigen Kunden mit dem zu erwartenden größten Interesse einschränkt. Somit spart die vorherige Kundensegmentierung die Kosten für wahrscheinlich unwirksame Marketing-Aktionen ein.

3.1.2 Formale Beschreibung von Akquiseplänen

Die Abb. 3.2 zeigt eine schematische Darstellung des Akquiseplans aus Abb. 3.1. Darin ist fol-

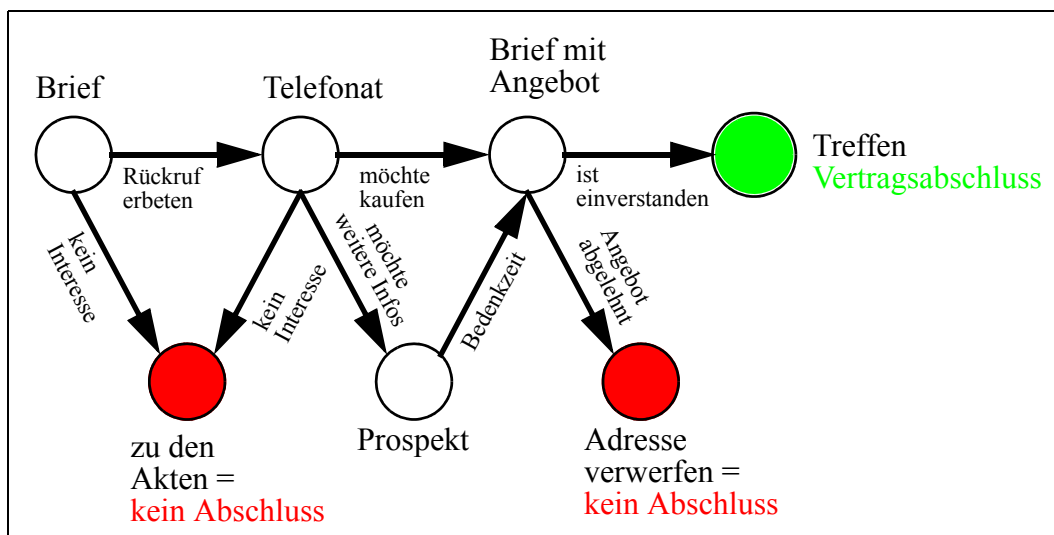


Abbildung 3.2: Formale Darstellung des Akquiseplans aus Abb. 3.1.

gendes zu erkennen:

- Ein Akquiseplan besteht aus *Knoten* (Aktionen des Anwenders, hier z. B. „Brief“) und *Kanten* (Reaktionen der Kunden, hier z. B. „möchte kaufen“) von einer Aktion zu einer nachfol-

genden Aktion. So kann ein Akquiseplan als *gerichteter, kreisfreier und zusammenhängender Graph* dargestellt werden.

- Kunden, die einen bestimmten Knoten durchlaufen, müssen nicht die gleiche „Vorgeschichte“ haben, das heißt vor diesem Knoten nicht gleiche Knoten durchlaufen haben. (Hier gibt es Kunden, die vor einem „Brief mit Angebot“ einen „Prospekt“ erhalten haben und Kunden, die keinen Prospekt erhalten haben.)
- Der Weg eines Kunden im Akquiseplan stellt auch graphentheoretisch einen *Weg* dar. Alle Kunden, die auf demselben Weg zu einem bestimmten Knoten gelangen, besitzen die gleiche „Vorgeschichte“. Abb. 3.3 zeigt einen Weg im Graphen aus Abb. 3.2.

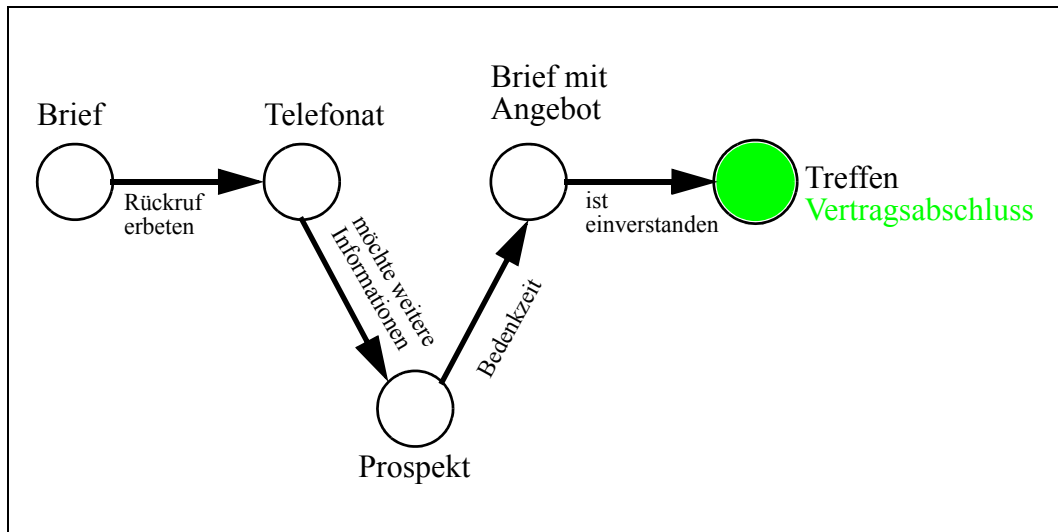


Abbildung 3.3: Ein Weg im Graphen aus Abb. 3.2.

- Alle Wege eines Kunden durch den Akquiseplan beginnen mit einer gemeinsamen ersten Aktion (hier: „Brief“) und enden entweder mit einem Vertragsabschluss (hier: „Treffen“) oder ohne Abschluss (hier: „Zu den Akten“, „Adresse verwerfen“), das heißt es gibt grundsätzlich eine Anfangsaktion und zwei Schlussaktionen.

Nach diesen Überlegungen kann jetzt genauer definiert werden:

Definition 3.1: *Akquiseplan.*

Sei \underline{A} eine endliche Menge von indizierten Akquiseaktionen, $\{A_1, A_2, \dots, A_N\}$ – die Knotenmenge – und \underline{R} eine Menge von Kunden-Reaktionen oder geordneten Aktionspaaren $\{(A_i, A_j) | A_i, A_j \in \underline{A}\}$ – die Kantenmenge. Seien $A_p, A_n \in \underline{A}$ zwei ausgezeichnete Aktionen, die einen Vertragsabschluss kennzeichnen

(„p“ = positiv) bzw. das Nichtzustandekommen eines Vertragsabschlusses („n“ = negativ).

Dann ist ein *Akquiseplan* ein gerichteter, schlichter³⁾, kreisfreier, eckenindizierter Graph $AP = (\underline{A}, \underline{R})$, der genau eine Quelle (= erste Aktion) $A_q \in \underline{A}$ und genau die beiden Senken A_p und A_n besitzt. \square

Wie aus Abb. 3.4 ersichtlich, ist der Akquiseplan aus Abb. 3.2 ein Akquiseplan gemäß Def. 3.1.

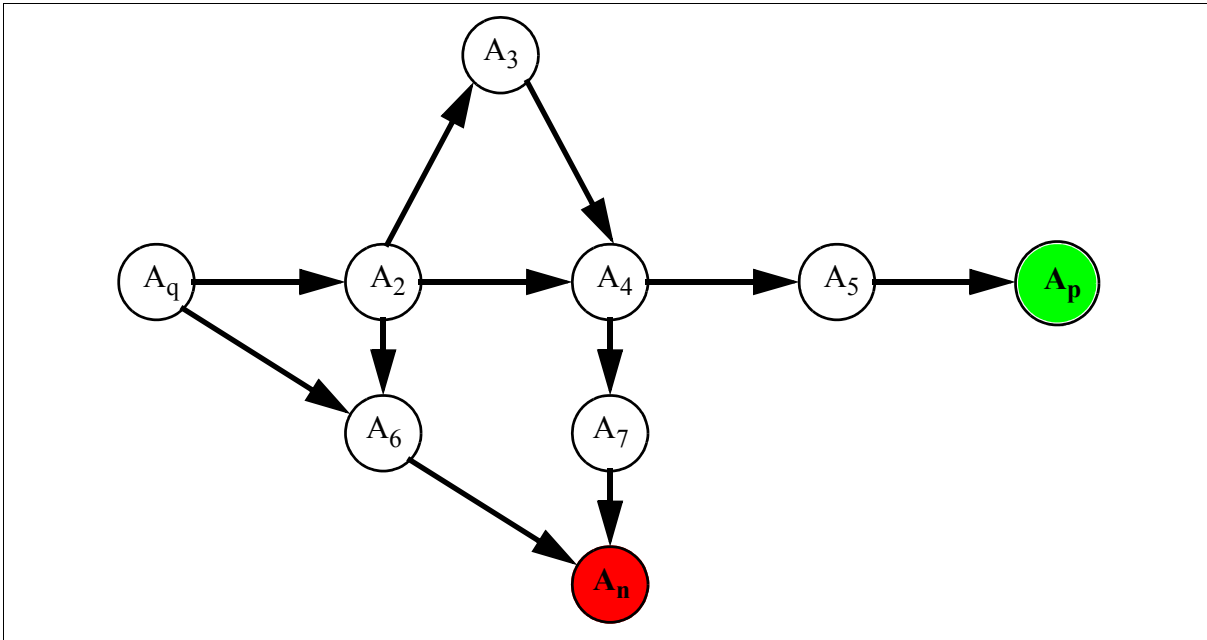


Abbildung 3.4: Darstellung des Akquiseplans aus Abb. 3.2 gemäß Def. 3.1: Anzahl der Knoten $N = 9$, $A_q = A_1$ = Brief, A_2 = Telefonat, A_3 = Prospekt, A_4 = Brief mit Angebot, A_5 = Treffen, A_6 = zu den Akten, A_7 = Adresse verwerfen, $A_n = A_8$ = kein Abschluss, $A_p = A_9$ = Vertragsabschluss.

Anmerkungen zu Def. 3.1

1. Ein Akquiseplan ist ein zusammenhängender Graph. Diese Eigenschaft folgt aus seiner Kreisfreiheit, und weil es genau eine Quelle gibt.
2. Für alle weiteren Betrachtungen sei vereinbart, dass ein Akquiseplan aus N Knoten bestehe, und dass $A_q = A_1$ sowie $A_p = A_N$ und $A_n = A_{N-1}$.
3. Jeder Knoten repräsentiert genau eine Aktion. Unterscheiden sich zwei Aktionen nicht in ihrer Form (z. B. zwei Briefe) aber in ihrem Inhalt (z. B. ein Brief mit einem Angebot, der zweite mit einem verbesserten Angebot), so werden die Aktionen auch im Akquiseplan unterschieden und durch unterschiedliche Knoten dargestellt.

3. Graphen ohne Schlingen und Mehrfachkanten werden *schlicht* genannt.

Definition 3.2: *Adjazenzmatrix eines Akquiseplans.*

Für jeden *Akquiseplan* $AP = (\underline{A}, \underline{R})$ mit der Knotenmenge $\underline{A} = \{A_1, \dots, A_N\}$ sei seine *Adjazenzmatrix* eine Matrix $A = (a_{ij}) \in \{0, 1\}^{N \times N}$ mit den Eigenschaften:

$$a_{ij} := \begin{cases} 1 & \text{für } (A_i, A_j) \in \underline{R} \\ 0 & \text{für } (A_i, A_j) \notin \underline{R} \end{cases}$$

für $i, j = 1, \dots, N$ [1]. □

Mit Ausnahme der Indizes 1, $N-1$ und N dienen die Indizes nur der unterscheidbaren Kennzeichnung der Aktionen und beziehen sich nicht auf die Position einer Aktion im Akquiseplan. Es wird sich aber im Folgenden als vorteilhaft erweisen, wenn die Aktionen so indiziert werden, dass die Adjazenzmatrix des Akquiseplans eine Block-Dreiecksgestalt annimmt, so dass von einer Aktion A_i , $i = 1, \dots, N-2$ nur Kanten zu Aktionen A_j mit größerem Index, das heißt $j > i$ verlaufen. Dass dies möglich ist, sagt der folgende Satz aus.

Satz 3.1: Die Adjazenzmatrix $A = (a_{ij})$, $A \in \{0, 1\}^{N \times N}$ des Akquiseplans $(\underline{A}, \underline{R})$, mit $N = |\underline{A}|$ kann durch Umindizierung der Knoten (Aktionen) $A_i \in \underline{A}$

in eine obere Block-Dreiecksmatrix der Gestalt $A' = (a'_{ij}) = \begin{pmatrix} \text{R} & \text{T} \\ \text{0} & \text{Z} \end{pmatrix}$ überführt

werden⁴⁾, wobei R eine obere Dreiecksmatrix in $\{0, 1\}^{(N-2) \times (N-2)}$ mit $\text{diag}R = 0$ und Z eine Nullmatrix in $\{0, 1\}^{2 \times 2}$ ist.

Beweis: Aus der Festlegung $A_p = A_N$ und $A_n = A_{N-1}$ folgt, dass A' die Gestalt $\begin{pmatrix} \text{R}' & \text{T}' \\ \text{0} & \text{Z}' \end{pmatrix}$

mit $Z \in \{0, 1\}^{2 \times 2}$ besitzt. Z ist eine Nullmatrix, weil A_{N-1} und A_N Senken des Akquiseplans sind. Laut Def. 3.1 ist der Graph $(\underline{A}, \underline{R})$ schlicht, besitzt also keine Schlingen, und deshalb ist $\text{diag}R' = 0$. Somit bleibt zu zeigen, dass R' durch Umindizierung in eine obere Dreiecksgestalt überführt werden kann.

Zu jeder Aktion A_i , $i = 2, \dots, N-2$ führt von der Quelle A_1 mindestens ein Kantenzug. Wird die Länge eines Kantenzugs als die Anzahl seiner Kanten definiert, dann kann jeder Aktion A_i , $i = 2, \dots, N-2$ die Länge des längsten Kantenzugs als Abstand $d(A_i)$ der Aktion A_i von der Quelle A_1 zugeordnet werden. Nun wird eine Neuindizierung

4. das heißt es gibt eine Permutationsmatrix $Q \in \{0, 1\}^{N \times N}$, so dass $A' = Q^T \cdot A \cdot Q$.

$$\nu : \{2, 3, \dots, N-2\} \rightarrow \{2, 3, \dots, N-2\}, A_i \rightarrow A'_{i'} = \nu(i)$$

der Aktionen durchgeführt, bei der die Indizes, ausgehend von A_1 so vergeben werden, dass $\nu(i) > \nu(j)$ für $d(A_i) > d(A_j)$ gilt. Unter den Aktionen mit gleichem Abstand, können die Indizes beliebig verteilt werden.

Nun ist zu zeigen, dass $a'_{i'j'} = 0$ für $j' \leq i'$ und alle $1 \leq i', j' \leq N-2$. Für $j' = i'$ folgt die Behauptung aus der Schlichkeit des Graphen. Für $j' < i'$ werde angenommen, das Gegenteil sei wahr, das heißt es gibt mindestens ein $a'_{i'j'} = 1$ mit $\nu(j) = j' < i' = \nu(i)$. Dann gibt es eine Kante $(A'_{\nu(i)}, A'_{\nu(j)})$ mit $\nu(j) < \nu(i)$. Weil $A'_{\nu(j)}$ Endpunkt der Kante ist, gilt gemäß obiger Definition des Abstands einer Aktion: $d(A'_{\nu(j)}) \geq d(A'_{\nu(i)}) + 1$. Nach den o. g. Indizierungsregeln folgt daraus $\nu(j) \geq \nu(i)$, was im Widerspruch zur Annahme $\nu(j) < \nu(i)$ steht. Also ist die Annahme falsch und somit $a'_{i'j'} = 0$ für $j' \leq i'$ und alle $1 \leq i', j' \leq N-2$. ■

Anmerkung zu Satz 3.1

Die Adjazenzmatrix eines Akquiseplans $AP = (\underline{A}, \underline{R})$ beschreibt eine *teilweise Ordnung* auf der Knotenmenge \underline{A} bezüglich der irreflexiven, asymmetrischen und transitiven „ist Vorgänger von“-Relation⁵⁾. Durch eine Umindizierung mit den beschriebenen Eigenschaften wird die teilweise Ordnung in eine lineare Ordnung eingebettet. Dies wird auch als *topologisches Sortieren* bezeichnet [169]. Ein alternativer konstruktiver Beweis wird in [63] gegeben.

Beispiel 3.1: Adjazenzmatrix für den Akquiseplan in Abb. 3.4. (Eine Umindizierung ist nicht notwendig, um die gewünschte Gestalt als obere Block-Dreiecksmatrix zu erreichen.)

$$\begin{array}{c} \begin{array}{c} 1 \\ \vdots \\ N-2 \\ \hline N \end{array} \begin{pmatrix} \begin{array}{cccccc|cc} 1 & & & & & & N-2 & N \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \end{pmatrix} \end{array}$$

3.1.3 Stärken und Schwächen von Akquiseplänen

Das Erstellen von Akquiseplänen mit dem Software-Werkzeug *AkquiSys* dient vor allem der Planung, Durchführung und Kontrolle von Marketingkampagnen für eine größere Zielgruppe.

5. Weil der Akquiseplan schlicht ist, kann die „ist Vorgänger von“-Relation nicht auf zwei identische Knoten angewendet werden. Somit ist sie nicht reflexiv und darum liegt keine Halbordnung vor.

Damit die Einfachheit des in Abb. 3.1 gezeigten Beispiels eines Akquiseplans nicht darüber hinwegtäuscht, dass diese Aufgaben nicht trivial sind, zeigt Abb. 3.5 den Akquiseplan eines Versicherungsagenten. Bei der Bearbeitung mehrerer hundert Kunden ist es offensichtlich nicht mehr einfach, den Überblick über den Stand der Marketingkampagne und die als nächstes zu bearbeitenden Kunden zu behalten. *AkquiSys* unterstützt den Akquisiteur durch

- die Berechnung des Aufwands an Kosten und Zeit für die Durchführung der Aktionen,
- so genannte „ToDo-Listen“ für Aufgaben, die der Akquisiteur als nächstes auszuführen hat [57].

Solche Informationen sind die Grundlage für die manuelle Auswertung von Akquiseplänen und ihre – wie auch immer definierte – manuelle Optimierung. Eine automatische Auswertung ist jedoch nicht möglich, und selbst der zeitliche Verlauf von Erlösen, Kosten und der Anzahl der in einer bestimmten Zeitspanne mit einer bestimmten Aktion bearbeiteten Kunden kann nicht angezeigt werden. Schließlich kann das Software-Werkzeug die Güte einzelner Aktionen oder des ganzen Akquiseplans nicht bewerten. Durch die mathematische Modellierung von Akquiseplänen wird in den folgenden Abschnitten die Grundlage zur Erweiterung der Funktionalität gelegt.

3.2 Mathematische Modellierung von Akquiseplänen

In diesem Abschnitt werden Akquisepläne so modelliert, dass der zeitliche Verlauf einer Marketingkampagne durch Umsetzung des mathematischen Modells in ein Rechnerprogramm simuliert werden kann. Das mathematische Modell und die Simulation ermöglichen die Bereitstellung folgender Informationen:

- Inanspruchnahme von Aktionen und Akquisewegen, insbesondere Angabe, welcher Teil der Kunden zu einem bestimmten Zeitpunkt bei Aktion A_i zur Bearbeitung erwartet wird; damit kann der Akquisiteur Auslastungsspitzen erkennen und eine Kapazitätsplanung durchführen.
- Bewertung von Aktionen, Akquisewegen und des ganzen Akquiseplans durch ökonomische Kenngrößen (Kosten, Einnahmen, Gewinn, Rendite); die Bewertung kann z. B. zur Beurteilung des Abbruchs einer Marketingkampagne dienen.
- Zeitlicher Verlauf der ökonomischen Kenngrößen; er gestattet Rückschlüsse auf den Einsatz der finanziellen Mittel und die wirtschaftlich sinnvolle Durchführung der Kundenakquisition.

Daraus ergibt sich, dass das mathematische Modell so beschaffen sein muss, dass Akquisepläne sowohl statisch als auch dynamisch betrachtet werden können.

- Bei der statischen Betrachtung eines Akquiseplans wird angenommen, dass sich alle Vorgänge einer auf dem Akquiseplan beruhenden Marketingkampagne innerhalb eines gegebenen Zeitraums vollständig vollziehen, ohne dass ihre Entwicklung in dieser Zeitspanne betrachtet wird. Bezüglich des Akquiseplans in Abb. 3.5 heißt das z. B., dass nach einer gewissen Zeit ein bestimmter Teil aller Kunden auf die Aktion A_5 so reagiert hat, dass Aktion A_6 folgt und der übrige Teil so, dass Aktion A_9 folgt.

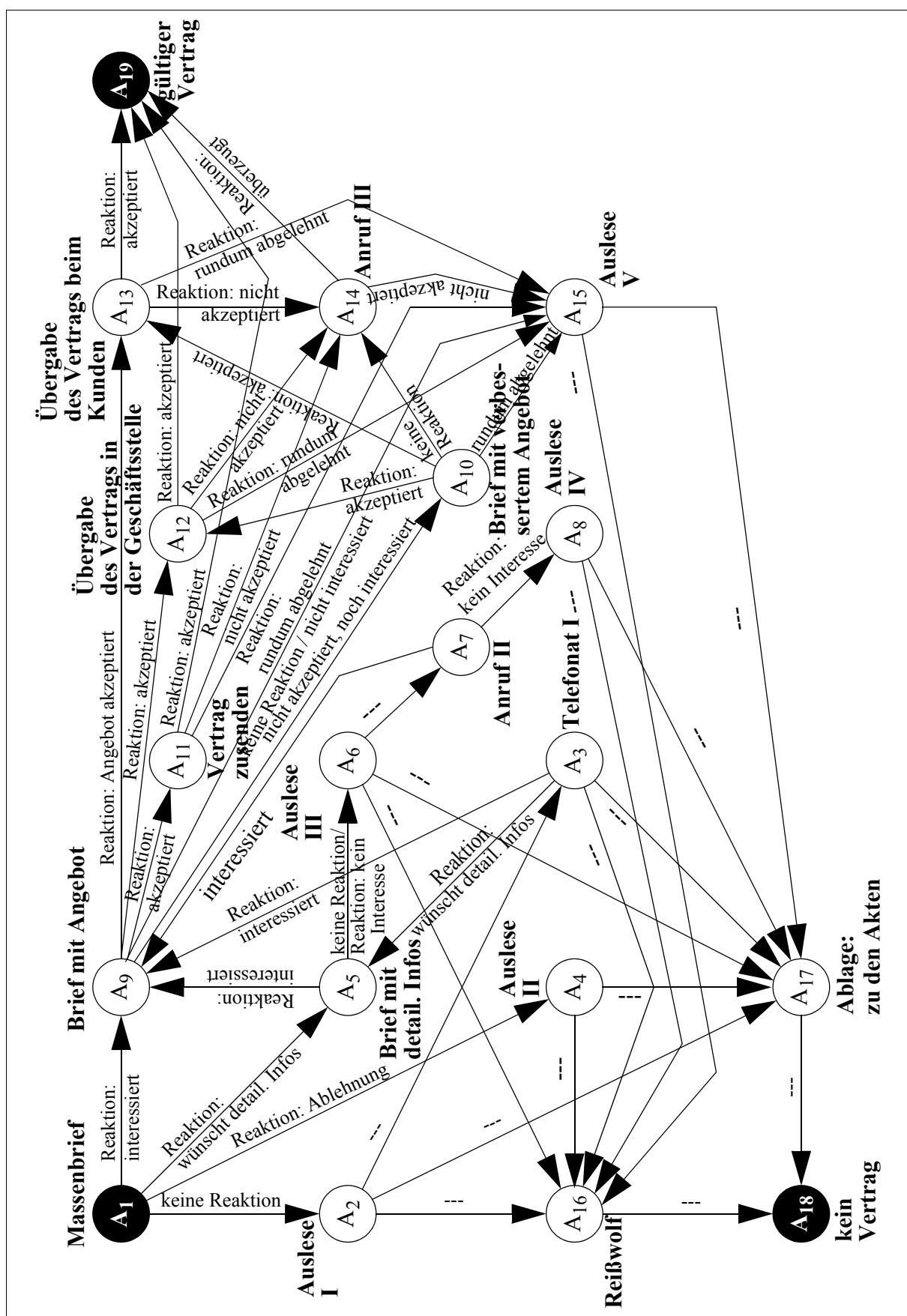


Abbildung 3.5: Beispiel: Akquiseplan eines Versicherungsagenten.

- Bei der dynamischen Betrachtung eines Akquiseplans wird die zeitliche Verteilung der Kundenaktionen (gemäß einer beliebigen zeitlichen Verteilung) mit betrachtet.

Weil üblicherweise eine Marketingkampagne nicht mit allen potenziellen Kunden gleichzeitig gestartet wird⁶⁾, kann dann auch die aufeinanderfolgende Zuführung neuer Kunden in den Akquiseplan simuliert werden.

Die Grundlage des mathematischen Modells ist die iterative Berechnung der Wahrscheinlichkeit $p_i(t)$, mit der sich ein Kunde zu einem gegebenen Zeitpunkt t , $t = 0, 1, \dots$ in einem gegebenen Knoten $A_i \in \underline{A}$, $i = 1, \dots, N$ des Akquiseplans $(\underline{A}, \underline{R})$ befindet. Das mathematische Modell wird in vier Stufen eingeführt:

1. Formale Beschreibung des Akquiseplans,
2. Einführung eines Wahrscheinlichkeitsraumes,
3. Definition von Aufenthalts- und Übergangswahrscheinlichkeiten in und zwischen den Knoten,
4. Einführung ökonomischer Größen.

Eine formale Beschreibung von Akquiseplänen ist bereits in Abschnitt 3.1.2 mit der Def. 3.1 erfolgt. Alle weiteren Stufen werden in den drei folgenden Abschnitten 3.2.1 – 3.2.3 eingeführt.

Anmerkungen zu Annahmen und Bezeichnungen

1. Ein Akquiseplan legt eine Marketingkampagne fest. Für jeden Kunden wird festgelegt, welche Marketing-Aktionen angewendet werden sollen. Durch seine Reaktion auf Marketing-Aktionen steuert jeder Kunde, welche Aktion als nächstes angewendet wird. Für jede Aktion seien Reaktionen so festgelegt, dass jeder Kunde auf einen Folgeknoten wechselt. Dies kann z. B. durch implizite Reaktionen, wie „keine Reaktion innerhalb 14 Tagen“ erreicht werden. So folgt jeder Kunde im Akquiseplan einem Pfad von der ersten Aktion A_1 bis zur Aktion A_{N-1} bzw. A_N . Der Anschaulichkeit halber, und um nicht zwischen dem Akquiseplan und einer Marketingkampagne auf Grundlage des Akquiseplans unterscheiden zu müssen, folgt der weitere Text der Sprechweise, dass jeder Kunde den Akquiseplan *durchläuft*. Wenn eine Marketing-Aktion auf einen Kunden angewandt wird, dann *befindet* sich der Kunde im entsprechenden Knoten, wenn der Kunde darauf reagiert, dann *geht* er auf einen Folgeknoten *über*.
2. Für die zeitliche Beschreibung der Marketingkampagne wird von einem diskreten Zeitraster $t = 0, 1, \dots$ ausgegangen bzw. von Zeitintervallen zwischen diesen Zeitpunkten. Dabei wird angenommen, dass alle Kunden zwischen zwei aufeinander folgenden Zeitpunkten höchstens einen Knotenwechsel vollziehen. Mit dieser Annahme betritt ein Kunde genau dann einen Knoten, wenn er auf die Aktion des Knotens, in welchem er sich zuletzt befunden hat, reagiert. Zugleich verlässt er den vorigen Knoten. Diese Betrachtungsweise setzt voraus, dass eine neue Aktion sofort nach einer Kundenreaktion ausgeführt wird, das heißt Aktionen benötigen höchstens einen Zeitschritt. Dagegen wird angenommen, dass eine

6. Die Ursache dafür können z. B. Kapazitätsbeschränkungen sein.

Kundenreaktion eine gewisse Zeit dauern kann. Während dieser Zeit *wartet* der Kunde im Knoten.

3. Für den Rest der Arbeit wird ein Akquiseplan $AP = (\underline{A}, \underline{R})$ mit $\underline{A} = \{A_1, \dots, A_N\}$ vorausgesetzt, für dessen Knoten gilt: $\forall (A_i, A_j) \in \underline{R} : j > i$ (vgl. Satz 3.1).

Für die folgenden Definitionen wird weiter abstrahiert und nur noch Knotenfolgen betrachtet. Eine Knotenfolge ist ein Zufallsereignis und für seine möglichen Realisierungen im Graphen des Akquiseplans können Wahrscheinlichkeiten angegeben werden.

3.2.1 Wahrscheinlichkeitsraum

In diesem Abschnitt wird ein Wahrscheinlichkeitsraum eingeführt, um Zufallsgrößen genau definieren zu können.

Definition 3.3: *Mögliche Knotenfolge.*

Gegeben sei der Akquiseplan $AP = (\underline{A}, \underline{R})$. Eine *mögliche Knotenfolge* $(a_t)_{t \geq 0}$ des Akquiseplans ist eine Folge aus $\{1, \dots, N\}$ mit folgenden Eigenschaften:

1. $(a_t)_{t \geq 0}$ ist schwach monoton wachsend, das heißt für beliebige $t_1, t_2 \in \mathbb{N} \cup \{0\}$ mit $t_1 < t_2$ gilt $a_{t_1} \leq a_{t_2}$.
2. $a_0 = 1$
3. $\exists t > 0 : (a_t = N - 1) \vee (a_t = N)$
4. Für alle $t > t'$ mit $(a_{t'} = N - 1) \vee (a_{t'} = N)$ gilt $(a_t = N - 1) \vee (a_t = N)$.
5. Für alle $t \geq 0$ mit $a_t \neq a_{t+1}$ gilt $(A_{a_t}, A_{a_{t+1}}) \in \underline{R}$.

Der Index $t = 0, 1, \dots$ bezeichnet diskrete Zeitpunkte (z. B. Tage oder Wochen) ab Beginn der Akquisition. \square

Anmerkung

Es gibt unendlich viele verschiedene *mögliche Knotenfolgen*.

Definition 3.4: *Merkmalsraum, Ereignisfeld, Ereignis, Wahrscheinlichkeitsverteilung, Wahrscheinlichkeitsraum.*

Gegeben sei der Akquiseplan AP . Der *Merkmalsraum* M des Akquiseplans ist die Menge aller möglichen Knotenfolgen gemäß Def. 3.3. Das *Ereignisfeld* E des Akquiseplans ist die Potenzmenge des Merkmalsraumes M . Seine Elemente heißen *Ereignisse*. Das Ereignisfeld enthält insbesondere die Ereignisse M und \emptyset (leere Menge). Die *Wahrscheinlichkeitsverteilung* W des Akquiseplans auf dem Ereignisfeld E ist eine Funktion $W : E \rightarrow [0, 1]$ mit folgenden Eigenschaften [156]:

1. $\forall e \in E : 0 \leq W\{e\} \leq 1$,
2. $W\{M\} = 1$,
3. Für jede Folge von Ereignissen $\{e_1, e_2, \dots\}$, $e_i \in E$ mit $e_i \cap e_j = \emptyset$ für $i \neq j$ gilt

$$W\left\{\bigcup_{i=1}^{\infty} e_i\right\} = \sum_{i=1}^{\infty} W\{e_i\}.$$

Der *Wahrscheinlichkeitsraum* des Akquiseplans AP ist das Tripel (M, E, W) aus seinem Merkmalsraum M , dem zugehörigen Ereignisfeld E und der Wahrscheinlichkeitsverteilung W ⁷⁾. □

Anmerkungen

1. Die Definition der Wahrscheinlichkeitsverteilung W setzt voraus, dass E bezüglich Durchschnitts- und Vereinigungsbildung abgeschlossen ist. Das wird dadurch gewährleistet, dass Ereignisfeld E die Potenzmenge von M ist.
2. Die Elemente der Merkmalsmenge sind einelementige Mengen von Knotenfolgen, $\{(a_t)_{t \geq 0}\}$. Diese Mengen gehören auch zum Ereignisraum und werden als Elementarereignisse bezeichnet [74]. Der Einfachheit halber werden sie mit $(a_t)_{t \geq 0}$ bezeichnet.

Definition 3.5: Ereignis „ein Kunde befindet sich zum Zeitpunkt t im Knoten i “.

Zu einem Akquiseplan AP mit dem Wahrscheinlichkeitsraum (M, E, W) ist das Ereignis $A(i, t) =$ „ein Kunde befindet sich zum Zeitpunkt t im Knoten i “ die Menge aller möglichen Knotenfolgen, deren t -tes Folgenglied gleich i ist, das heißt

$$A(i, t) := \bigcup_{(a_\tau)_{\tau \geq 0} \in M \mid a_t = i} \{(a_\tau)\} . \quad (3.1)$$

Das Ereignis $\bar{A}(i, t) =$ „ein Kunde befindet sich zum Zeitpunkt t nicht im Knoten i “ ist das Komplement von $A(i, t)$, das heißt

$$\bar{A}(i, t) := M \setminus A(i, t) = \bigcup_{(a_\tau)_{\tau \geq 0} \in M \mid a_t \neq i} \{(a_\tau)\} . \quad (3.2)$$

Ferner wird aus formalen Gründen $\bar{A}(i, -1)$ eingeführt⁸⁾, so dass gilt

$$A(i, 0) \cap \bar{A}(i, -1) := A(i, 0) . \quad (3.3)$$

□

7. In der Literatur ist auch der Begriff „Wahrscheinlichkeitsmaß“ gebräuchlich, der die gleiche Bedeutung hat [74].

8. Diese Konvention wird im Beweis von Hilfssatz 3.2 benötigt.

Anmerkungen

1. Weil nach Def. 3.5 $A(i, t)$ und $\bar{A}(i, t)$ Elemente des Ereignisfelds E sind, liegen auch alle Durchschnitte und Vereinigungen von Ereignissen $A(i, t)$ und $\bar{A}(i, t)$, $i = 1, \dots, N$, $t = 0, 1, \dots$ in E , so dass ihre Wahrscheinlichkeiten definiert sind.
2. Da nach Def. 3.3 $a_0 = 1$ ist, folgt aus (3.3):

$$\bar{A}(i, -1) := A(i, 0) = \begin{cases} M & \text{für } i = 1 \\ \emptyset & \text{für } i > 1 \end{cases}.$$

Mit Hilfe der Ereignisse $A(i, t)$ und $\bar{A}(i, t)$, $i = 1, \dots, N$, $t = 0, 1, \dots$ des Akquiseplans AP mit dem Wahrscheinlichkeitsraum (M, E, W) werden im nächsten Abschnitt Wahrscheinlichkeiten definiert, mit denen sich der zeitliche Ablauf der Kundengewinnung mit dem Akquiseplan AP beschreiben lässt. Ihr Endresultat ist der Satz 3.12 über die iterative Berechnung der Präsenzwahrscheinlichkeiten, die die Grundlage des in Abb. 3.1 angegebenen Algorithmus ist.

3.2.2 Wahrscheinlichkeiten und ihre Zusammenhänge

Für die folgenden Definitionen wird ein Akquiseplan AP mit einem Wahrscheinlichkeitsraum (M, E, W) und gemäß Def. 3.5 definierten Ereignissen $A(i, t)$ angenommen.

3.2.2.1 Präsenzwahrscheinlichkeiten

Die Präsenzwahrscheinlichkeiten geben an, mit welcher Wahrscheinlichkeit sich ein Kunde in einem bestimmten Knoten eines Akquiseplans befindet.

Definition 3.6: *Präsenzwahrscheinlichkeit, Vektor der Präsenzwahrscheinlichkeiten.*

$p_i(t) = W\{A(i, t)\}$ ist die Wahrscheinlichkeit, dass sich ein Kunde zum Zeitpunkt t im Knoten A_i befindet, das heißt die *Präsenzwahrscheinlichkeit* für Knoten A_i , $i = 1, \dots, N$ zum Zeitpunkt t , $t = 0, 1, \dots$. Die Präsenzwahrscheinlichkeiten werden im *Vektor der Präsenzwahrscheinlichkeiten* zum Zeitpunkt t , $p(t) = (p_1(t), \dots, p_N(t))$ zusammengefasst. \square

Anmerkungen

1. (M, E, W) ist ein diskreter Wahrscheinlichkeitsraum. Wegen (3.1) gilt daher für die Präsenzwahrscheinlichkeit [155]:

$$p_i(t) = \sum_{(a_\tau)_{\tau \geq 0} \in M \mid a_t = i} W\{(a_\tau)\}. \quad (3.4)$$

2. Die Präsenzwahrscheinlichkeiten sind grundlegende Größen für die Simulation des Akquiseprozesses im Abschnitt 3.3 und für das im Abschnitt 3.4 formulierte Optimierungsproblem. Während ein Akquiseplan von einem Fachmann mit Praxiserfahrung leicht erstellt werden kann, ist es schwierig, für alle Knoten A_i und alle Zeitpunkte t die Präsenzwahrscheinlichkeiten anzugeben. Das liegt vor allem daran, dass ein Mensch im zeitlichen Verlauf einer Akquise weniger die Verteilung der Kunden auf die einzelnen Knoten zu den Zeitpunkten $t = 0, 1, \dots$ wahrnimmt, sondern vor allem, *wie* die Kunden, auf die eine Aktion angewendet wurde, reagieren und *wie schnell* sie reagieren. Dieses Verhalten wird hier durch die *Wahrscheinlichkeit des Wechsels nach Ankunft im Knoten*, p_{ij} (Def. 3.8) und durch die *zeitliche Punktwahrscheinlichkeit des Knotenwechsels*, $\varphi_{ij}(t)$ (Def. 3.9) beschrieben. Für den realistischen Akquiseplan aus Abb. 3.5 konnten diese Größen mit geringem Aufwand von einem Fachmann geschätzt werden⁹⁾. Mit den folgenden Definitionen und Sätzen wird gezeigt, wie aus p_{ij} und $\varphi_{ij}(t)$ die Präsenzwahrscheinlichkeiten $p_i(t)$ berechnet werden können.
3. Alternativ zur Schätzung durch einen Fachmann könnten die Größen p_{ij} und $\varphi_{ij}(t)$ bei mengenmäßig und qualitativ ausreichender Datenbasis auch aus Daten geschätzt werden, z. B. bei mehrmaliger Benutzung des gleichen Akquiseplans.

Um die Kunden in einem Knoten nach ihrer Aufenthaltsdauer im Knoten unterscheiden zu können, wird die *bedingte Wahrscheinlichkeit des Knoteneintritts* eingeführt.

Definition 3.7: *Bedingte Wahrscheinlichkeit des Knoteneintritts.*

Die Wahrscheinlichkeit

$$\omega_i^t(\tau) = W\{A(i, t-\tau) \cap \bar{A}(i, t-\tau-1) \mid A(i, t)\}$$

ist die *bedingte Wahrscheinlichkeit des Knoteneintritts*, das heißt die Wahrscheinlichkeit, dass ein Kunde, der sich zum Zeitpunkt t , $t = 0, 1, \dots$ im Knoten A_i befindet, im Zeitraum $(t-\tau-1, t-\tau]$ den Knoten erreicht hat. \square

Die beiden folgenden Hilfssätze beschreiben Zusammenhänge, die sich weiter unten als nützlich erweisen werden.

Hilfssatz 3.2: Für die Ereignisse $A(i, t)$, $i = 1, \dots, N$, $t = 0, 1, \dots$ gelten folgende Zusammenhänge:

$$W\left\{\bigcup_{i=1}^N A(i, t)\right\} = 1 \text{ für alle } t = 0, 1, \dots, \quad (3.5)$$

9. Dem Kundenverhalten, das durch die angeführten Größen beschrieben wird, liegen natürlich objektive Ursachen zugrunde, die mithilfe besonderer Verfahren (z. B. der mikrogeografischen Segmentierung) ermittelt werden können. Aus Platzgründen kann in dieser Arbeit nicht auf sie eingegangen werden.

$$W\{A(i, t_1) \cap A(i, t_2) \cap A(i, t_3)\} = W\{A(i, t_1) \cap A(i, t_3)\} \\ \text{für beliebige } 0 \leq t_1 \leq t_2 \leq t_3, \quad (3.6)$$

$$W\left\{\bigcup_{\tau=0}^t [A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)] \mid A(i, t)\right\} = 1. \quad (3.7)$$

Beweis:

zu (3.5): Mit Def. 3.5 folgt

$$\bigcup_{i=1}^N A(i, t) = \bigcup_{i=1}^N \left[\bigcup_{(a_\tau)_{\tau \geq 0} \in M \mid a_t = i} \{(a_\tau)\} \right] = \bigcup_{(a_\tau)_{\tau \geq 0} \in M} \{(a_\tau)\} = M.$$

Also ist

$$W\left\{\bigcup_{i=1}^N A(i, t)\right\} = W\{M\} = 1.$$

zu (3.6): Gemäß den Definitionen 3.3 und 3.5 ist $A(i, t_1) \cap A(i, t_2) \cap A(i, t_3)$ die Menge aller möglichen Knotenfolgen $(a_t)_{t \geq 0}$, für die gilt $a_{t_1} = a_{t_2} = a_{t_3} = i$. Daraus folgt, dass diese Menge in $A(i, t_1) \cap A(i, t_3)$ enthalten ist. Andererseits ist $A(i, t_1) \cap A(i, t_3)$ die Menge aller möglichen Knotenfolgen $(a_t)_{t \geq 0}$, für die gilt $a_{t_1} = a_{t_3} = i$. Weil Knotenfolgen gemäß Def. 3.3 schwach monoton wachsen, kann für ein t_2 mit $0 \leq t_1 \leq t_2 \leq t_3$ nur $a_{t_2} = i$ gelten. Also ist $A(i, t_1) \cap A(i, t_3)$ in $A(i, t_1) \cap A(i, t_2) \cap A(i, t_3)$ enthalten. Somit sind die Mengen $A(i, t_1) \cap A(i, t_3)$ und $A(i, t_1) \cap A(i, t_2) \cap A(i, t_3)$ gleich und deshalb gilt

$$W\{A(i, t_1) \cap A(i, t_2) \cap A(i, t_3)\} = W\{A(i, t_1) \cap A(i, t_3)\}.$$

zu (3.7): Aus der Definition einer bedingten Wahrscheinlichkeit folgt

$$W\left\{\bigcup_{\tau=0}^t [A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)] \mid A(i, t)\right\} \\ = \frac{W\left\{\left[\bigcup_{\tau=0}^t [A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)]\right] \cap A(i, t)\right\}}{W\{A(i, t)\}}.$$

Jetzt ist zu zeigen, dass

$$\bigcup_{\tau=0}^t [A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)] \cap A(i, t) = A(i, t) \text{ für } i = 1, \dots, N, t = 0, 1, \dots, \quad (3.8)$$

das heißt dass gilt:

$$\bigcup_{\tau=0}^t [A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)] \supseteq A(i, t). \quad (3.9)$$

Nach den Definitionen 3.3 und 3.5 (und unter Beachtung der Konvention in (3.3)) ist die linke Seite dieser Mengenrelation die Menge aller möglichen Knotenfolgen $(a_\vartheta)_{\vartheta \geq 0}$, für die entweder $a_0 = i$ oder $a_\tau = i$ und $a_{\tau-1} < i$ für ein $\tau \in \{1, \dots, t\}$ ist, also der Knotenfolgen, die den Knotenindex i spätestens im t -ten Folgenglied erreichen. Diese Menge umfasst offensichtlich alle möglichen Knotenfolgen $(a_\vartheta)_{\vartheta \geq 0}$ mit $a_t = i$. ■

Anmerkungen zu Hilfssatz 3.2

Der Hilfssatz sagt Folgendes aus:

1. Jeder Kunde befindet sich zu jedem Zeitpunkt fast sicher¹⁰⁾ in einem der Knoten des Akquiseplans, das heißt bei jedem Kunden wird zu jedem Zeitpunkt auf seine Reaktion auf eine Aktion gewartet.
2. Kein Kunde kann den Knoten eines Akquiseplans verlassen und danach wieder den gleichen Knoten erreichen, das heißt fast sicher keine Aktion kann zweimal auf den gleichen Kunden angewandt werden.
3. Jeder Kunde, der sich zu einem bestimmten Zeitpunkt in einem Knoten (außer dem Quellknoten) des Akquiseplans befindet, ist zuvor fast sicher von einem anderen Knoten zu diesem Knoten gewechselt.

Hilfssatz 3.3: Für die Wahrscheinlichkeiten $\omega_i^t(\tau)$ und $p_i(t)$, $i = 1, \dots, N$, $t = 0, 1, \dots$, $\tau = 0, \dots, t$ gelten die folgenden Zusammenhänge:

$$\sum_{\tau=0}^t \omega_i^t(\tau) = W \left\{ \bigcup_{\tau=0}^t A(i, t-\tau) \cap \bar{A}(i, t-\tau-1) \mid A(i, t) \right\} = 1, \quad (3.10)$$

$$\sum_{\tau=0}^t p_i(t) \omega_i^t(\tau) = p_i(t) \text{ für } t = 0, 1, \dots. \quad (3.11)$$

Beweis:

Die Aussage (3.10) folgt unmittelbar aus (3.7) in Hilfssatz 3.2 und (3.11) direkt aus (3.10). ■

3.2.2.2 Übergangswahrscheinlichkeiten

Neben den Präsenzwahrscheinlichkeiten sind für die Simulation einer Akquise die Wahrscheinlichkeiten für den Übergang eines Kunden von einem Knoten zu einem anderen Knoten wichtig, weil der Knotenübergang der Reaktion des Kunden auf eine Aktion entspricht.

10. „fast sicher“, das heißt mit Wahrscheinlichkeit 1

Nachfolgend werden zwei Übergangswahrscheinlichkeiten eingeführt, die folgende Fragen beantworten:

- Mit welcher Wahrscheinlichkeit kann von einem Kunden, auf den *gerade eben* eine Marketing-Aktion angewandt wurde, eine *bestimmte* Reaktion erwartet werden? Die Frage richtet sich also nicht nach dem Zeitpunkt der Reaktion, sondern ob der Kunde *irgendwann* eine *bestimmte* Reaktion zeigt. Indirekt kann der Zeitpunkt wichtig sein, dann nämlich, wenn nach einer festgelegten Zeit „keine Reaktion“ festgestellt wird. Die Antwort darauf geben die *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten* (Def. 3.8), die im zeitinvarianten Fall eine gewöhnliche Übergangsmatrix liefern (siehe Anmerkungen auf Seite 61).
- Mit welcher Wahrscheinlichkeit kann vom Kunden *bis zum nächsten Zeittakt* eine *bestimmte* Reaktion erwartet werden? Ob ein Kunde bis zum nächsten Zeittakt reagiert, hängt sicherlich davon ab, *wann* die Marketing-Aktion auf ihn angewandt worden ist, das heißt wie lange er sich schon „im Knoten“ befindet bzw. dort *wartet*. Die Antwort darauf geben die *Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten* (Def. 3.10).

Definition 3.8: *Wahrscheinlichkeit des Wechsels nach Ankunft im Knoten.*

Die bedingte Wahrscheinlichkeit

$$p_{ij}^t := W \left\{ \bigcup_{\tau=0}^{\infty} (A(j, t+\tau+1) \cap A(i, t+\tau)) \mid A(i, t) \cap \bar{A}(i, t-1) \right\} \quad (3.12)$$

ist die *Wahrscheinlichkeit des Wechsels nach Ankunft im Knoten*, das heißt die Wahrscheinlichkeit, dass ein Kunde, der im Zeitraum $(t-1, t]$ den Knoten A_i erreicht hat, irgendwann später zum Knoten A_j wechselt, für beliebige $t = 0, 1, \dots$ und beliebige $i, j \in \{1, \dots, N\}$. \square

Anmerkung

Sofern p_{ij}^t zeitinvariant ist, also unabhängig von t , wird nur p_{ij} geschrieben.

Definition 3.9: *Zeitliche Punktwahrscheinlichkeit des Knotenwechsels, zeitliche Verteilungsfunktion des Knotenwechsels.*

Die Funktion

$$\varphi_{ij}(\tau) : \mathbb{N}_0 \rightarrow [0, 1]$$

$$\text{mit } \varphi_{ij}(\tau) := W \left\{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \right. \\ \left. \mid \bigcup_{\vartheta=0}^{\infty} A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1) \right\} \quad (3.13)$$

für $\tau = 0, 1, \dots$ und beliebiges $t = 0, 1, \dots$ ist die *zeitliche Punktwahrscheinlichkeit des Knotenwechsels*, das heißt $\varphi_{ij}(\tau)$ ist die Wahrscheinlichkeit, dass ein

Kunde, der den Knoten A_i , $i = 1, \dots, N$ zu einem beliebigen Zeitpunkt t erreicht hat, im Zeitintervall $(t + \tau, t + \tau + 1]$ zum Knoten A_j , $j = 1, \dots, N$ wechselt.

Die *zeitliche Verteilungsfunktion des Knotenwechsels* ist

$$\phi_{ij}(t) := \sum_{\tau=0}^t \varphi_{ij}(\tau) \text{ für } t = 0, 1, \dots \quad (3.14)$$

□

Eine Rechtfertigung der Bezeichnung *zeitliche Punktwahrscheinlichkeit des Knotenwechsels* gibt der folgende Hilfssatz.

Hilfssatz 3.4: Für die zeitliche Punktwahrscheinlichkeit des Knotenwechsels

$\varphi_{ij}(\tau)$ gemäß Def. 3.9 gilt: $\sum_{\tau=0}^{\infty} \varphi_{ij}(\tau) = 1$.

Beweis: Gemäß Def. 3.9 gilt für jeden Zeitpunkt t :

$$\begin{aligned} \sum_{\tau=0}^{\infty} \varphi_{ij}(\tau) &= \sum_{\tau=0}^{\infty} \frac{W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}}{W\left\{\bigcup_{\vartheta=0}^{\infty} A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1)\right\}} \\ &= \sum_{\tau=0}^{\infty} \frac{W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}}{\sum_{\vartheta=0}^{\infty} W\{A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1)\}} \\ &= \frac{\sum_{\tau=0}^{\infty} W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}}{\sum_{\tau=0}^{\infty} W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}} = 1 \end{aligned}$$

■

Satz 3.5: Die Wahrscheinlichkeit, dass ein Kunde, der im Zeitraum $(t-1, t]$ den Knoten A_i erreicht hat, im Zeitraum $(t + \tau, t + \tau + 1]$ zum Knoten A_j wechselt, beträgt

$$p_{ij}^t \cdot \varphi_{ij}(\tau)$$

für alle $t = 0, 1, \dots$ und alle $\tau = 0, 1, \dots$

Beweis: Aus (3.13) folgt:

$$\varphi_{ij}(\tau) = \frac{W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}}{W\left\{\bigcup_{\vartheta=0}^{\infty} A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1)\right\}}.$$

Aus (3.12) folgt:

$$p_{ij}^t = \frac{W\left\{\bigcup_{\tau=0}^{\infty} (A(j, t+\tau+1) \cap A(i, t+\tau)) \cap A(i, t) \cap \bar{A}(i, t-1)\right\}}{W\{A(i, t) \cap \bar{A}(i, t-1)\}}.$$

Damit ist

$$\begin{aligned} p_{ij}^t \cdot \varphi_{ij}(\tau) &= \frac{W\{A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)\}}{W\{A(i, t) \cap \bar{A}(i, t-1)\}} \\ &= W\{A(j, t+\tau+1) \cap A(i, t+\tau) \mid A(i, t) \cap \bar{A}(i, t-1)\}. \end{aligned} \quad (3.15)$$

■

Die Größen p_{ij}^t und $\varphi_{ij}(\tau)$ sind nicht nur von praktischer Bedeutung (siehe Anmerkung auf Seite 54), sondern auch theoretisch wichtig:

Satz 3.6: Durch Angabe von p_{ij}^t und $\varphi_{ij}(t)$ gemäß der Definitionen 3.8 und 3.9 für $t = 0, 1, \dots$ und alle $i, j = 1, \dots, N$ ist die Wahrscheinlichkeitsverteilung W des Wahrscheinlichkeitsraumes aus Def. 3.4 eindeutig festgelegt.

Beweis: Sei $(a_t)_{t \geq 0}$ eine beliebige *mögliche Knotenfolge* gemäß Def. 3.3. O. B. d. A. besitze die Folge $n \leq N-1$ unterschiedliche Folgenglieder. Die Zeitpunkte $0 =: t_0 < t_1 < \dots < t_{n-1}$ seien so gewählt, dass $a_{t_{i-1}} < a_{t_i}$ und $a_{t_{i-1}} = \dots = a_{t_i-1}$ jeweils für $i = 1, \dots, n-1$ und für $t_{n-1} < t$ ist $a_{t_{n-1}} = a_t$. Das heißt, die Knotenfolge bleibt vor Zeitpunkt t_i für $(t_i - 1) - t_{i-1}$ Zeitschritte konstant und im Intervall $(t_{i-1}, t_i]$ ändert sich das Folgenglied für $i = 1, \dots, n-1$. Damit beträgt die Wahrscheinlichkeit der Knotenfolge $(a_t)_{t \geq 0}$:

$$\begin{aligned} W\{(a_t)_{t \geq 0}\} &= W\{A(a_{t_0}, t_0)\} \cdot W\{A(a_{t_1}, t_1) \cap A(a_{t_0}, t_1-1) \mid A(a_{t_0}, t_0)\} \\ &\cdot W\{A(a_{t_2}, t_2) \cap A(a_{t_1}, t_2-1) \mid A(a_{t_1}, t_1) \cap \bar{A}(a_{t_1}, t_1-1)\} \cdot \dots \\ &\cdot W\{A(a_{t_{n-1}}, t_{n-1}) \cap A(a_{t_{n-2}}, t_{n-1}-1) \mid A(a_{t_{n-2}}, t_{n-2}) \cap \bar{A}(a_{t_{n-2}}, t_{n-2}-1)\} \end{aligned} \quad (3.16)$$

Wegen $t_0 = 0$ und $A_{t_0} = 1$ ist $W\{A(a_{t_0}, t_0)\} = 1$. Und mit der Konvention (3.3) folgt für (3.16):

$$W\{(a_t)_{t \geq 0}\} = \prod_{i=1}^{n-1} W\{A(a_{t_i}, t_i) \cap A(a_{t_{i-1}}, t_i-1) \mid A(a_{t_{i-1}}, t_{i-1}) \cap \bar{A}(a_{t_{i-1}}, t_{i-1}-1)\}.$$

Und mit (3.15) ist schließlich

$$W\{(a_t)_{t \geq 0}\} = \prod_{i=1}^{n-1} p_{a_{t_{i-1}} a_{t_i}}^{t_{i-1}} \cdot \varphi_{a_{t_{i-1}} a_{t_i}}(t_i - t_{i-1} - 1).$$

Damit ist die Wahrscheinlichkeit der *möglichen Knotenfolge* $(a_t)_{t \geq 0}$ durch die *Übergangswahrscheinlichkeiten nach Ankunft im Knoten*, $p_{a_{t_{i-1}} a_{t_i}}^{t_{i-1}}$ und durch die *zeitlichen Punktwahrscheinlichkeit*, $\varphi_{a_{t_{i-1}} a_{t_i}}(t_i - t_{i-1} - 1)$, jeweils für $i = 1, \dots, n-1$ eindeutig gegeben. ■

Anmerkung zu Satz 3.6

Für einen gegebenen Akquiseplan AP ist die Festlegung des *Wahrscheinlichkeitsraumes* aus Def. 3.4 äquivalent zur Festlegung von *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten* p_{ij}^t und von *zeitlichen Punktwahrscheinlichkeiten des Knotenwechsels* $\varphi_{ij}(t)$.

Definition 3.10: *Wahrscheinlichkeit des Wechsels nach Wartezeit im Knoten.*

Die bedingte Wahrscheinlichkeit

$$\pi_{ij}^t(\tau) = W\{A(j, t+1) \mid A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}$$

ist die *Wahrscheinlichkeit des Wechsels nach Wartezeit im Knoten*, das heißt die Wahrscheinlichkeit, dass ein Kunde, der im Zeitraum $(t-\tau-1, t-\tau]$ den Knoten A_i , $i = 1, \dots, N$ erreicht hat, im Zeitintervall $(t, t+1]$ vom Knoten A_i zum Knoten A_j , $j = 1, \dots, N$ wechselt, wobei $t = 0, 1, \dots$ und $\tau = 0, \dots, t$.

Sofern $\pi_{ij}^t(\tau)$ zeitinvariant ist, also unabhängig von t , wird nur $\pi_{ij}(\tau)$ geschrieben. □

Anmerkungen

1. Für den Rest der Arbeit werden die Größen p_{ij}^t und $\varphi_{ij}^t(\tau)$ als zeitinvariant vorausgesetzt und – soweit die Angabe des Zeitpunktes t nicht notwendig ist – die Bezeichnungen p_{ij} bzw. $\varphi_{ij}(\tau)$ verwendet. Wenn diese Größen von einem Fachmann geschätzt werden müssen, dann ist ihre zeitabhängige Modellierung praktisch nicht möglich.
2. Werden $p_{N-1, N-1} = p_{N, N} := 1$ gesetzt und $p_{N-1, j} := 0$ für $j \neq N-1$ sowie $p_{N, j} := 0$ für $j \neq N$, jeweils mit $1 \leq j \leq N$, dann ist die Übergangsmatrix $P := (p_{ij}) \in [0, 1]^{N \times N}$ eine stochastische Matrix¹¹⁾. Weil die Adjazenzmatrix des Akquiseplans AP eine obere Block-Dreiecksmatrix ist (vgl. Satz 3.1), hat die Übergangsmatrix P ebenfalls Block-Dreiecksgestalt,

$$\begin{pmatrix} \text{R} & \text{T} \\ \text{0} & \text{E} \end{pmatrix},$$

wobei $R \in [0, 1]^{(N-2) \times (N-2)}$ eine obere Dreiecksmatrix mit $\text{diag}R = 0$, also insbesondere nilpotent ist, und $E \in [0, 1]^{2 \times 2}$ ist eine Einheitsmatrix.

3.2.2.3 Zusammenhang zwischen den Wahrscheinlichkeiten

Der abschließende Satz 3.12 in diesem Abschnitt erlaubt es, die Präsenzwahrscheinlichkeiten $p_i(t)$ zu allen Zeitpunkten zu berechnen. Die zunächst folgenden Sätze und Hilfssätze werden für seinen Beweis benötigt. Mit ihrer Hilfe wird der Zusammenhang zwischen den Präsenzwahrscheinlichkeiten aus Abschnitt 3.2.2.1 und den Übergangswahrscheinlichkeiten aus Abschnitt 3.2.2.2 hergestellt.

Hilfssatz 3.7: Für die Ereignisse $A(i, t)$, $i = 1, \dots, N-2$, $t = 0, 1, \dots$ gilt der Zusammenhang

$$A(i, t) = \bigcup_{j=i+1}^N \bigcup_{\vartheta=0}^{\infty} [A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t)] .$$

Beweis: Die linke Seite ist gleich der Menge möglicher Knotenfolgen $(a_{\vartheta})_{\vartheta \geq 0}$ mit $a_t = i$. Weil $i < N-1$ ist, müssen diese Folgen ab einem bestimmten Folgenglied einen Wert $j > i$ annehmen. Das sind alle möglichen Knotenfolgen $(a_{\vartheta})_{\vartheta \geq 0}$, deren Glieder den Wert i bis $a_{t+\vartheta}$, $\vartheta \geq 0$ beibehalten und für die $a_{t+\vartheta+1} > i$, das heißt die linke Menge ist gleich der rechten Menge. ■

Anmerkung zu Hilfssatz 3.7

Der Satz sagt aus, dass die *Anwesenheit* eines Kunden in einem Knoten (der nicht Endknoten ist) zugleich bedeutet, dass der Kunde *irgendwann* zu *irgend einem* anderen Knoten *wechseln* wird.

Satz 3.8: Für $t = 0, 1, \dots$ gilt für die Aufenthaltswahrscheinlichkeit im Knoten A_j zum Zeitpunkt $t+1$:

$$p_j(t+1) = \sum_{i=1}^N p_i(t) \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ij}^t(\tau) .$$

-
11. Eine Matrix $P = (p_{ij})_{i,j=1,\dots,N}$ ist eine stochastische Matrix, genau dann wenn alle $p_{ij} \geq 0$, $i, j = 1, \dots, N$ und jede ihrer Zeilensummen gleich eins ist, das heißt für jedes $i = 1, \dots, N$ gilt $\sum_{j=1}^N p_{ij} = 1$ [155].
-

Beweis: Es ist

$$\begin{aligned}
 & \sum_{i=1}^N p_i(t) \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ij}^t(\tau) \\
 &= \sum_{i=1}^N W\{A(i, t)\} \sum_{\tau=0}^t W\{A(i, t-\tau) \cap \bar{A}(i, t-\tau-1) \mid A(i, t)\} \\
 & \quad \cdot W\{A(j, t+1) \mid A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\} \\
 &= \sum_{i=1}^N \sum_{\tau=0}^t W\{A(j, t+1) \cap A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\} \\
 & \stackrel{\text{Disjunktheit}}{=} W\left\{A(j, t+1) \cap \bigcup_{i=1}^N \left[\bigcup_{\tau=0}^t (A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)) \right] \right\} \\
 & \stackrel{(3.8)}{=} W\left\{A(j, t+1) \cap \bigcup_{i=1}^N A(i, t)\right\} \\
 &= \left\{A(j, t+1) \cap \bigcup_{i=1}^j A(i, t)\right\} \cup \left\{A(j, t+1) \cap \bigcup_{i=j+1}^N A(i, t)\right\}.
 \end{aligned}$$

Weil die Ereignisse $A(j, t+1)$ und $A(i, t)$ über Knotenfolgen definiert sind und eine Knotenfolge schwach monoton wächst, ist der Durchschnitt $A(j, t+1) \cap A(i, t)$ für $i > j$ und damit die Menge in der rechten Klammer leer. Damit ist

$$\sum_{i=1}^N p_i(t) \sum_{\tau=0}^t \omega_i^t(t-\tau) \pi_{ij}^t(\tau) = W\left\{A(j, t+1) \cap \bigcup_{i=1}^N A(i, t)\right\}.$$

Zu dieser Menge gehören alle möglichen Knotenfolgen $(a_\emptyset)_{\emptyset \geq 0}$, für die $a_{t+1} = j$ und $a_t \leq j$. Dies sind nach Def. 3.3 alle möglichen Knotenfolgen $(a_\emptyset)_{\emptyset \geq 0}$ mit $a_{t+1} = j$. Somit

$$\text{ist } A(j, t+1) \cap \bigcup_{i=1}^j A(i, t) = A(j, t+1) \text{ und}$$

$$W\left\{A(j, t+1) \cap \bigcup_{i=1}^j A(i, t)\right\} = W\{A(j, t+1)\}. \quad \blacksquare$$

Dieses Ergebnis deutet eine mögliche rekursive Berechnung der Präsenzwahrscheinlichkeiten $p_i(t)$ an. Wenn ein Experte die Größen $\omega_i^t(\tau)$ und $\pi_{ij}^t(\tau)$ für alle $t = 0, 1, \dots$ und $\tau = 0, 1, \dots, t$ sowie $i, j \in \{1, \dots, N\}$ angeben könnte, würde Satz 3.8 ausreichen. Wie in den Anmerkungen zu Def. 3.6 angeführt wird, ist es für einen Fachmann einfacher, die Größen p_{ij} (Def. 3.8) und die zeitliche Punktwahrscheinlichkeit $\varphi_{ij}(t)$ (Def. 3.9) festzulegen. Deshalb wird in den folgenden Sätzen 3.9 – 3.11 gezeigt, wie die Wahrscheinlichkeiten $\omega_i^t(\tau)$ und $\pi_{ij}^t(\tau)$ aus p_{ij} und $\varphi_{ij}(t)$ berechnet werden können.

Satz 3.9: Für $\pi_{ij}^t(\tau)$ und p_{ij} , $t = 0, 1, \dots$, $\tau = 0, 1, \dots, t$, $i \neq j$ gelten folgende Zusammenhänge:

$$p_{ij} = 0 \Leftrightarrow \forall \tau > 0 : \pi_{ij}^{t+\tau}(\tau) = 0 ; \quad (3.17)$$

$$\pi_{ij}^{t+\tau}(\tau) = \frac{p_{ij} \varphi_{ij}(\tau)}{\sum_{k=i+1}^N p_{ik} \sum_{\vartheta=\tau}^{\infty} \varphi_{ik}(\vartheta)} \quad \text{für } 0 \leq i < N-1. \quad (3.18)$$

Beweis:

zu (3.17): Es gilt

$$\begin{aligned} p_{ij}^t &= W \left\{ \bigcup_{\tau=0}^{\infty} (A(j, t+\tau+1) \cap A(i, t+\tau)) \mid A(i, t) \cap \bar{A}(i, t-1) \right\} \\ &= \frac{W \left\{ \bigcup_{\tau=0}^{\infty} (A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1)) \right\}}{W \{ A(i, t) \cap \bar{A}(i, t-1) \}} \\ &= \frac{\sum_{\tau=0}^{\infty} W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{W \{ A(i, t) \cap \bar{A}(i, t-1) \}} \end{aligned}$$

und

$$\begin{aligned} \pi_{ij}^{t+\tau}(\tau) &= W \{ A(j, t+\tau+1) \mid A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \} \\ &= \frac{W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{W \{ A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}. \end{aligned}$$

Also ist $p_{ij}^t = 0 \Leftrightarrow \forall \tau : W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \} = 0$ für beliebige $t = 0, 1, \dots$.

Das ist genau dann der Fall, wenn $\pi_{ij}^{t+\tau}(\tau) = 0$. ■

zu (3.18): Es ist

$$\pi_{ij}^{t+\tau}(\tau) = \frac{W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{W \{ A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}} \neq 0. \quad (3.19)$$

Mit Hilfssatz 3.7 ist $A(i, t) = \bigcup_{j=i+1}^N \bigcup_{\vartheta=0}^{\infty} [A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t)]$. Daraus

folgt

$$W \{ A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}$$

$$\begin{aligned}
&= W \left\{ \bigcup_{j=i+1}^N \bigcup_{\vartheta=\tau}^{\infty} [A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1)] \right\} \\
&= \sum_{j=i+1}^N \sum_{\vartheta=\tau}^{\infty} W \{ A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1) \} .
\end{aligned}$$

Eingesetzt in (3.19) ergibt sich

$$\begin{aligned}
\pi_{ij}^{t+\tau}(\tau) &= \frac{W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{\sum_{k=i+1}^N \sum_{\vartheta=\tau}^{\infty} W \{ A(k, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1) \}} \\
&= \frac{W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{W \{ A(i, t) \cap \bar{A}(i, t-1) \}} \\
&= \frac{\sum_{j=i+1}^N \sum_{\vartheta=\tau}^{\infty} \frac{W \{ A(j, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1) \}}{W \{ A(i, t) \cap \bar{A}(i, t-1) \}}}{\sum_{k=i+1}^N \sum_{\vartheta=\tau}^{\infty} W \{ A(k, t+\vartheta+1) \cap A(i, t+\vartheta) \cap A(i, t) \cap \bar{A}(i, t-1) \}} \\
&= \frac{W \{ A(j, t+\tau+1) \cap A(i, t+\tau) \mid A(i, t) \cap \bar{A}(i, t-1) \}}{\sum_{k=i+1}^N \sum_{\vartheta=\tau}^{\infty} W \{ A(k, t+\vartheta+1) \cap A(i, t+\vartheta) \mid A(i, t) \cap \bar{A}(i, t-1) \}} \\
&= \frac{p_{ij} \varphi_{ij}(\tau)}{(3.15) \sum_{k=i+1}^N p_{ik} \sum_{\vartheta=\tau}^{\infty} \varphi_{ik}(\vartheta)} .
\end{aligned}$$

■

Anmerkung zu Satz 3.9

Gleichung 3.18 mag auf den ersten Blick verwirren, insbesondere scheinen sich die Wahrscheinlichkeiten auf der linken Seite und im Zähler des Terms auf der rechten Seite auf das gleiche Ereignis zu beziehen, so dass die Frage aufkommen könnte, wozu der Nenner auf der rechten Seite dient. Zum genauen Verständnis ist zu beachten, dass auf der linken Seite mit $\pi_{ij}^t(\tau)$ und auf der rechten Seite im Nenner mit $p_{ij} \varphi_{ij}(\tau)$ zwar bedingte Wahrscheinlichkeiten stehen, ihre Voraussetzungen aber verschieden sind. So setzt $\pi_{ij}^t(\tau)$ voraus, dass sich ein Kunde seit genau τ Zeitschritten im Knoten A_i befindet, während $p_{ij} \varphi_{ij}(\tau)$ nur voraussetzt, dass sich ein Kunde im Knoten A_i befindet. Damit beide Seiten übereinstimmen, muss $p_{ij} \varphi_{ij}(\tau)$ auf die Wahrscheinlichkeit, dass sich ein Kunde im Knoten A_i befindet und dort mindestens τ Zeitschritte verbleibt, bezogen werden. Wegen $i < N - 1$ ist dies die Wahrscheinlichkeit der Kunden, die zur Zeit t im Knoten A_i sind und nach τ Zeiteinheiten oder später zu irgendeinem anderen Knoten wechseln. Diese Wahrscheinlichkeit steht im Nenner auf der rechten Seite.

Satz 3.10: Für die Berechnung von $\pi_{ii}^t(\tau)$, $i = 1, \dots, N$ und $t = 0, 1, \dots$ gilt:

$$\pi_{ii}^t(\tau) = 1 - \sum_{j=i+1}^N \pi_{ij}^t(\tau) . \quad (3.20)$$

Beweis: Man zeigt, dass $\sum_{j=i}^N \pi_{ij}^t(\tau) = 1$. Nach Def. 3.10 ist für ein beliebiges $t \in \mathbb{N}_0$:

$$\begin{aligned} \sum_{j=i}^N \pi_{ij}^t(\tau) &= \sum_{k=j}^N W\{A(j, t+1) \mid A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\} \\ &= \sum_{j=i}^N \frac{W\{A(j, t+1) \cap A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}}{W\{A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}} \\ &= \frac{W\left\{\bigcup_{j=i}^N A(j, t+1) \cap A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\right\}}{W\{A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}} \\ &\stackrel{\text{Disjunktheit}}{=} \frac{W\{M \cap A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}}{W\{A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}} \\ &= \frac{W\{A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}}{W\{A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}} = 1 . \quad \blacksquare \end{aligned}$$

Satz 3.11: Für $p_i(t+1) \neq 0$ gilt

$$\omega_i^{t+1}(\tau) = \frac{p_i(t) \cdot \omega_i^t(\tau-1) \cdot \pi_{ii}^t(\tau-1)}{p_i(t+1)} , \quad (3.21)$$

für $t = 0, 1, \dots$ und $\tau = 1, \dots, t+1$.

Weiterhin ist

$$\omega_i^{t+1}(0) = 1 - \sum_{\tau=1}^{t+1} \omega_i^{t+1}(\tau) . \quad (3.22)$$

Beweis:

zu (3.21): Es ist zu zeigen:

$$p_i(t+1) \cdot \omega_i^{t+1}((t+1)-\tau) = p_i(t) \cdot \omega_i^t(t-(\tau-1)) \cdot \pi_{ii}^t(\tau-1) .$$

Aus den Definitionen 3.7 und 3.8 folgt für die linke Seite

$$\begin{aligned} p_i(t+1) \cdot \omega_i^{t+1}((t+1)-\tau) &= W\{A(i, t+1)\} \cdot W\{A(i, t+1-\tau) \cap \bar{A}(i, t-\tau) \mid A(i, t+1)\} \\ &= W\{A(i, t+1) \cap A(i, t+1-\tau) \cap \bar{A}(i, t-\tau)\} \end{aligned}$$

und für die rechte Seite folgt mit Def. 3.10

$$\begin{aligned}
p_i(t) \cdot \omega_i^t(t - (\tau - 1)) \cdot \pi_{ii}^t(\tau - 1) &= W\{A(i, t)\} \cdot W\{A(i, t + 1 - \tau) \cap \bar{A}(i, t - \tau) \mid A(i, t)\} \\
&\quad \cdot W\{A(i, t + 1) \mid A(i, t) \cap A(i, t - (\tau - 1)) \cap \bar{A}(i, t - \tau)\} \\
&= W\{A(i, t)\} \cdot \frac{W\{A(i, t + 1 - \tau) \cap \bar{A}(i, t - \tau) \cap A(i, t)\}}{W\{A(i, t)\}} \\
&\quad \cdot \frac{W\{A(i, t + 1) \cap A(i, t) \cap A(i, t - (\tau - 1)) \cap \bar{A}(i, t - \tau)\}}{W\{A(i, t) \cap A(i, t - (\tau - 1)) \cap \bar{A}(i, t - \tau)\}} \\
&\stackrel{(3.6)}{=} W\{A(i, t + 1) \cap A(i, t + 1 - \tau) \cap \bar{A}(i, t - \tau)\}. \quad \blacksquare
\end{aligned}$$

Satz 3.12: Die Vektoren der Präsenzwahrscheinlichkeiten $p(t)$ können für $t = 0, 1, \dots$ iterativ durch Matrizenmultiplikation berechnet werden. Hierzu sei

$$P_\omega(t + 1) := (\psi_{ij}^{t+1})_{i,j=1,\dots,N} \quad (3.23)$$

mit

$$\psi_{ij}^{t+1} := \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ij}^t(\tau). \quad (3.24)$$

Dann gilt

$$p(t + 1) = p(t) P_\omega(t + 1) = p(0) \prod_{\tau=1}^{t+1} P_\omega(\tau) \quad (3.25)$$

und für jedes $t = 0, 1, \dots$ ist $P_\omega(t + 1)$ eine stochastische Matrix¹²⁾.

Beweis: Die Gleichungen (3.24) und (3.25) folgen unmittelbar aus Satz 3.8. In (3.24) können die Werte für die $\pi_{ij}^t(\tau)$ gemäß Satz 3.9 und Satz 3.10 für gegebene p_{ij} und φ_{ij} berechnet werden. Die Werte für die $\omega_i^t(\tau)$ lassen sich iterativ gemäß Satz 3.11 berechnen.

Für beliebiges $i \in \{1, \dots, N\}$ und beliebiges $t = 0, 1, \dots$ beträgt die Zeilensumme von $P_\omega(t + 1)$:

$$\sum_{j=1}^N \psi_{ij}^{t+1} = \sum_{j=1}^N \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ij}^t(\tau) = \sum_{\tau=0}^t \omega_i^t(\tau) \left[\sum_{j=1}^N \pi_{ij}^t(\tau) \right] \stackrel{(3.20)}{=} \sum_{\tau=0}^t \omega_i^t(\tau) \stackrel{(3.22)}{=} 1.$$

Damit ist $P_\omega(t + 1)$ für beliebiges $t = 0, 1, \dots$ eine stochastische Matrix. ■

12. Der Index ω in der Bezeichnung der Matrix dient der Unterscheidung von der Matrix $P := (p_{ij}) \in [0, 1]^{N \times N}$ der Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten gemäß Def. 3.8. Vergleiche auch Anmerkung 2 zu Def. 3.10. Er weist darauf hin, dass in die Berechnung der ψ_{ij}^{t+1} die bedingten Wahrscheinlichkeiten des Knoteneintritts $\omega_i^t(\tau)$ eingehen.

In Abschnitt 3.4 zeigt Alg. 3.1 den Algorithmus zur Berechnung der Präsenzwahrscheinlichkeiten.

3.2.2.4 Dauer der Akquise

Für die praktische Nutzung des hier entwickelten Modells ist es wichtig, dass Akquisen von endlicher Dauer sind.

Definition 3.11: *Dauer einer Akquise.*

Die *Dauer einer Akquise* mit einem Akquiseplan AP und dem Wahrscheinlichkeitsraum (M, E, W) ist gleich der Anzahl der Zeitschritte, nach denen sich erstmals alle Kunden mit Wahrscheinlichkeit eins in einem der Endknoten A_{N-1} oder A_N befinden. \square

Anmerkung

Wenn für die Präsenzwahrscheinlichkeiten in den Endpunkten des Akquiseplans AP gilt: $p_{N-1}(t'-1) + p_N(t'-1) < 1$ und $p_{N-1}(t') + p_N(t') = 1$, dann bezeichnet t' die *Dauer der Akquise*.

Im Folgenden wird gezeigt, dass die Dauer einer Akquise von der Eigenschaft des Trägers der zeitlichen Punktwahrscheinlichkeiten $\varphi_{ij}(t)$, $i, j = 1, \dots, N$ abhängt.

Definition 3.12: *Endlicher Träger einer zeitlichen Punktwahrscheinlichkeit.*

Eine zeitliche Punktwahrscheinlichkeit $\varphi_{ij}(t)$ gemäß Def. 3.9 besitzt einen endlichen Träger genau dann, wenn es ein $t' > 0$ gibt, so dass gilt:

$$\varphi_{ij}(t) \begin{cases} \geq 0 & \text{für } 0 \leq t \leq t_{ij}' \\ = 0 & \text{für } t_{ij}' < t \end{cases}.$$

\square

Weiter unten zeigt Satz 3.16, dass die Dauer einer Akquise endlich ist, wenn alle zeitlichen Punktwahrscheinlichkeiten $\varphi_{ij}(t)$, $i, j = 1, \dots, N$ einen endlichen Träger besitzen. Für den Beweis des Satzes sind zunächst eine Definition und mehrere Hilfssätze einzuführen. Die folgende Definition wird in Kapitel 4 wieder aufgegriffen.

Definition 3.13: *(vollständige) Menge der Vorgängerknoten/-aktionen, (vollständige) Menge der Nachfolgerknoten/-aktionen eines Knotens/einer Aktion.*

Sei AP ein Akquiseplan und $A_{AP} = (a_{ij}) \in \{0, 1\}^{N \times N}$ seine Adjazenzmatrix. Dann ist

$$\text{Vorg}(j) := \{i \in \{1, \dots, N\} : a_{ij} = 1\}$$

die Menge der Vorgängerknoten/-aktionen des Knotens/der Aktion j und

$$\text{Nachf}(j) := \{k \in \{1, \dots, N\} : a_{jk} = 1\}$$

die Menge der Nachfolgerknoten/-aktionen des Knotens/der Aktion j , jeweils für $j = 1, \dots, N$. Die vollständigen Mengen der Vorgänger- bzw. Nachfolgerknoten/-aktionen entsprechen den Mengen der Vorgänger- bzw. Nachfolgerknoten/-aktionen unter Einschluss des betrachteten Knotens, das heißt:

$$\overline{\text{Vorg}}(i) := \text{Vorg}(i) \cup \{i\} \text{ bzw. } \overline{\text{Nachf}}(i) := \text{Nachf}(i) \cup \{i\}.$$

□

Die folgenden Hilfssätzen machen Aussagen über wichtige Eigenschaften der Übergangsmatrizen $P_\omega(t+1)$ aus Satz 3.12.

Hilfssatz 3.13: Sei $P \in [0, 1]^{N \times N}$ eine reellwertige Matrix mit Block-Dreiecksgestalt

$$P = \begin{pmatrix} R & T \\ 0 & E \end{pmatrix}, \quad (3.26)$$

wobei $R \in \mathbb{R}^{(N-2) \times (N-2)}$ eine obere Dreiecksmatrix mit $\text{diag}R = 0$, und $E \in \mathbb{R}^{2 \times 2}$ eine Einheitsmatrix ist. Dann gilt für $k \geq N-2$:

$$P^k = \begin{pmatrix} 0 & T' \\ 0 & E \end{pmatrix}.$$

Beweis: Sei

$$\begin{pmatrix} R(k) & T(k) \\ 0 & E(k) \end{pmatrix} := P^k.$$

Aus (3.26) ist ersichtlich, dass $R(k) = R^k$. Da R nilpotent ist gilt $R^k = 0$ für $k \geq N-2$. Für $E(k)$ gilt offensichtlich: $E(k) = E^k = E$. ■

Hilfssatz 3.14: Gegeben sei ein Akquiseplan AP und der Wahrscheinlichkeitsraum (M, E, W) . Dann besitzt die Matrix $P_\omega(t+1)$, $t = 0, 1, \dots$ gemäß Satz 3.12 Block-Dreiecksgestalt

$$P_{\omega}(t+1) = \begin{pmatrix} R & T \\ 0 & E \end{pmatrix}, \quad (3.27)$$

wobei $R \in [0, 1]^{(N-2) \times (N-2)}$ eine obere Dreiecksmatrix mit $\text{diag} R \geq 0$ und $E \in [0, 1]^{2 \times 2}$ eine Einheitsmatrix ist.

Beweis: Mit (3.17) in Satz 3.9 folgt für $i, j \in \{1, \dots, N\}$ mit $i \neq j$: $[p_{ij} = 0] \Rightarrow [\pi_{ij}(\tau) = 0]$ für alle $\tau \geq 0$. Mit (3.24) in Satz 3.12 folgt daraus $\psi_{ij}^{t+1} = 0$ für $i \neq j$ und alle $t = 0, 1, \dots$. Damit ist die Eigenschaft von R gezeigt sowie, dass $\psi_{N-1, N}^{t+1} = \psi_{N, N-1}^{t+1} = 0$. Da $P_{\omega}(t+1)$ eine stochastische Matrix ist, folgt $\psi_{N-1, N-1}^{t+1} = \psi_{N, N}^{t+1} = 1$, das heißt E ist die 2×2 -Einheitsmatrix. ■

Hilfssatz 3.15: Gegeben sei ein Akquiseplan AP , der Wahrscheinlichkeitsraum (M, E, W) und $P_{\omega}(t+1)$, die zugehörige Übergangsmatrix im Schritt $t+1$, $t = 0, 1, \dots$ aus Satz 3.12 mit der Gestalt (3.27). Wenn die zeitlichen Punktwahrscheinlichkeiten $\varphi_{ij}(\tau)$, $i < j$, $i, j \in \{1, \dots, N\}$ einen endlichen Träger besitzen, dann verschwinden die Diagonalelemente in R nach endlich vielen Zeitschritten. □

Beweis: O. B. d. A. gelte für jede zeitliche Punktwahrscheinlichkeit $\varphi_{ij}(\tau_{ij}') > 0$ für $i < j$ und $i, j \in \{1, \dots, N\}$. Sei

$$\tau_i' := \max_{j \in \text{Nachf}(i)} \tau_{ij}' \quad (3.28)$$

für $i = 1, \dots, N-2$ die maximale Aufenthaltsdauer eines Kunden im Knoten A_i mit Wahrscheinlichkeit eins. Dann gibt es für jedes $i \in \{1, \dots, N-2\}$ mindestens ein $j \in \{2, \dots, N\}$ mit $j > i$, so dass $\varphi_{ij}(\tau_i') > 0$.

Weiterhin sei für jeden Knoten $\tilde{\tau}_j := \max_{i \in \text{Vorg}(j)} (\tilde{\tau}_i + \tau_{ij}')$ für $j > 1$ und $\tilde{\tau}_1 := 0$ die Ankunftszeit des letzten Kunden mit Wahrscheinlichkeit eins.

Im Folgenden wird ein beliebiger Knoten A_i , $i \in \{1, \dots, N-2\}$ betrachtet. Nach den Sätzen 3.9 und 3.10 gilt für die Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten:

$$\pi_{ii}^{t+\tau}(\tau) \cdot \sum_{j=i+1}^N p_{ij} \sum_{\vartheta=\tau}^{\infty} \varphi_{ij}(\vartheta) = \sum_{j=i+1}^N p_{ij} \sum_{\vartheta=\tau}^{\infty} \varphi_{ij}(\vartheta) - \sum_{j=i+1}^N p_{ij} \varphi_{ij}(\tau). \quad (3.29)$$

Nach Voraussetzung gilt für alle $j > i$, $j \in \{1, \dots, N\}$: $\varphi_{ij}(\tau_i' + 1) = 0$. Damit folgt aus (3.29) für $\tau = \tau_i'$:

$$\pi_{ii}^{t+\tau_i'}(\tau_i') \cdot \sum_{j=i+1}^N p_{ij} \varphi_{ij}(\tau_i') = \sum_{j=i+1}^N p_{ij} \varphi_{ij}(\tau_i') - \sum_{j=i+1}^N p_{ij} \varphi_{ij}(\tau_i'). \quad (3.30)$$

Nach (3.28) sind die Summen ungleich null, so dass nach Division durch $\sum_{j=i+1}^N p_{ij} \varphi_{ij}(\tau_i')$ auf beiden Seiten von (3.30) gilt:

$$\pi_{ii}^{t+\tau_i'}(\tau_i') = 1 - 1 = 0. \quad (3.31)$$

Gemäß (3.24) ist

$$\psi_{ii}^{t+1} := \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ii}^t(\tau). \quad (3.32)$$

Auf der rechten Seite kann die Indexmenge wegen (3.31) nach oben durch $\min\{t, \tau_i' - 1\}$ beschränkt werden. Im Knoten A_i kommen die Kunden mit Wahrscheinlichkeit eins spätestens im Zeitschritt $\tilde{\tau}_i$ an. Somit ist zum Zeitpunkt t mit Wahrscheinlichkeit eins kein Kunde länger als $t - \tilde{\tau}_i$ Zeitschritte im Knoten A_i . Daher kann in (3.32) die Indexmenge der Summe nach unten durch $t - \tilde{\tau}_i$ beschränkt werden. Somit gilt für (3.32):

$$\psi_{ii}^{t+1} := \sum_{\tau=t-\tilde{\tau}_i}^{\min\{t, \tau_i'-1\}} \omega_i^t(\tau) \pi_{ii}^t(\tau).$$

Mit $t = \tau_i' + \tilde{\tau}_i$ folgt damit

$$\psi_{ii}^{t+1} = \sum_{\tau=\tau_i'}^{\tau_i'-1} \omega_i^t(\tau) \pi_{ii}^t(\tau) = 0.$$

Damit ist die Aussage des Satzes gezeigt. ■

Satz 3.16: Gegeben sei ein Akquiseplan AP und der Wahrscheinlichkeitsraum (M, E, W) . Wenn die zeitlichen Punktwahrscheinlichkeiten $\varphi_{ij}(\tau)$, $i < j$, $i, j \in \{1, \dots, N\}$ einen endlichen Träger besitzen, dann ist die Dauer der Akquise endlich.

Beweis: Sei $P_\omega(t+1)$ die zugehörige Übergangsmatrix im Schritt $t+1$, $t = 0, 1, \dots$ aus Satz 3.12 mit der Gestalt (3.27). Hilfssatz 3.15 besagt, dass es ein $t' \in \mathbb{N}$ gibt, so dass für die Teilmatrix R von $P_\omega(t'+1)$ gilt: $\text{diag} R = 0$. Sei

$$\begin{pmatrix} R(t'+k) & T' \\ 0 & E \end{pmatrix} := \prod_{k=1}^{N-2} P_\omega(t'+k).$$

Dann folgt mit Hilfssatz 3.13: $R(t' + k) = 0$. Und aufgrund der besonderen Gestalt der Übergangsmatrizen ist

$$\prod_{\tau=1}^{t'+N-2} P_{\omega}(\tau) = \left[\prod_{k=1}^{N-2} P_{\omega}(t' + k) \right] \cdot \left[\prod_{\tau=1}^{t'} P_{\omega}(\tau) \right] = \left[\prod_{k=1}^{N-2} P_{\omega}(t' + k) \right].$$

Mit $p(0) = (1, 0, \dots, 0)$ folgt mit Satz 3.12:

$$\begin{aligned} p(t' + N - 2) &= (0, \dots, 0, p_{N-1}(t' + N - 2), p_N(t' + N - 2)) \\ &= (0, \dots, 0, \psi_{1,N-1}^{t'+N-2}, \psi_{1,N}^{t'+N-2}), \end{aligned}$$

und da $P_{\omega}(t' + N - 2)$ eine stochastische Matrix ist, folgt

$$p_{N-1}(t' + N - 2) + p_N(t' + N - 2) = 1,$$

das heißt, die Akquise ist zum Zeitpunkt $t' + N - 2$ beendet. ■

3.2.3 Ökonomische Größen

In der letzten Stufe der mathematischen Modellbildung werden ökonomische Größen, wie Kosten, Ertrag, Gewinn und Rendite eingeführt. Dabei werden Kosten formal als negative Erträge betrachtet. Die Begriffe Kosten, Ertrag, Gewinn und Rendite sind in der Betriebswirtschaftslehre, insbesondere in der Kostenrechnung mit bestimmten aber kontextabhängigen Inhalten belegt [135]. Weil sich die vorliegende Arbeit nicht auf eine bestimmte Branche konzentriert, werden diese Begriffe hier sehr allgemein gebraucht:

- *Kosten* werden Aktionen zugeordnet und bezeichnen (die vom Anwender festgelegten) Ausgaben, die bei der Ausführung der Aktion entstehen (z. B. Briefporto). (Es handelt sich um Durchschnittswerte über Kunden, Zeiten usw.)
- *Ertrag* wird der Aktion A_N zugeordnet und bezeichnet (die vom Anwender festgelegten) Einnahmen beim Kauf- oder Vertragsabschluss (z. B. Provision für einen Versicherungsabschluss). (Es handelt sich um Durchschnittswerte; ggf. sind damit auch Einnahmen in der Zukunft, z. B. periodische Versicherungsprovisionen gemeint.)
- Mit *Gewinn* wird die Differenz aus Ertrag und Kosten bezeichnet, also eine Größe für den wirtschaftlichen Erfolg der Akquisition.
- *Rendite* ist der Quotient Gewinn/Kosten. Er bezeichnet sozusagen die Verzinsung des eingesetzten Kapitals und ist ebenfalls eine Größe für den wirtschaftlichen Erfolg der Akquisition.

Die genannten Größen beziehen sich auf einzelne (durchschnittliche) Kunden. Bezüglich einer Kundenmenge wird jeweils von *Gesamtkosten*, *-ertrag*, *-gewinn* und *-rendite* gesprochen. Sie sollen in zeitlicher Abhängigkeit für den ganzen Akquiseplan, einzelne Wege und Knoten, sowie für Kunden mit positivem Geschäftsabschluss und für Kunden, mit denen kein Geschäft zustande kommt, berechnet werden. Einen Überblick der Größen und ihrer Bezeichnungen gibt Tabelle 3.1. Als Grundlage zur Berechnung obiger Größen wird eine *Ertragsfunktion* eingeführt:

Größe	Knoten A_j	Akquiseplan AP	Weg w^a	Kunden mit Abschluss	Kunden ohne Abschluss
Kosten	Kosten gem. Def. 3.14: k_j ; Kosten im Zeitpunkt t : $k_j(t)$; Kosten im Zeitraum $[0, t]$: $k_j[0, t]$ (3.33), (3.34) (nur A_1, \dots, A_{N-2})	Kosten im Zeitpunkt t : $k_{AP}(t)$; Kosten im Zeitraum $[0, t]$: $k_{AP}[0, t]$ (3.36), (3.37)	Kosten im Zeitpunkt t : $k_w(t)$; Kosten im Zeitraum $[0, t]$: $k_w[0, t]$ (nur Kosten, auch wenn w in A_N endet)	Kosten im Zeitpunkt t : $k_N(t)$; Kosten im Zeitraum $[0, t]$: $k_N[0, t]$ (Aggregation über alle w , die in A_N enden – nur Kosten)	Kosten im Zeitpunkt t : $k_{N-1}(t)$; Kosten im Zeitraum $[0, t]$: $k_{N-1}[0, t]$ (Aggregation über alle w , die in A_{N-1} enden)
Ertrag	Ertrag gem. Def. 3.14: e_N ; Ertrag im Zeitpunkt t : $e_N(t)$; Ertrag im Zeitraum $[0, t]$: $e_N[0, t]$ (3.35) (nur A_N)	Ertrag im Zeitpunkt t : $e_{AP}(t)$; Ertrag im Zeitraum $[0, t]$: $e_{AP}[0, t]$ (3.38), (3.39)	Ertrag im Zeitpunkt t : $e_w(t)$; Ertrag im Zeitraum $[0, t]$: $e_w[0, t]$ (nur w , die in A_N enden)	siehe Spalte <i>Knoten A_j</i>	Kunden ohne Abschluss erzeugen keinen Ertrag

Tabelle 3.1: Übersicht von ökonomischen Größen, die in einer Simulation des Akquise-Prozesses berechnet werden könnten. Alle Größen sind Erwartungswerte für einen Kunden. Die Tabellenzeilen beziehen sich auf die verschiedenen Größen, die Tabellenspalten auf den betrachteten Aspekt der Akquise, der jeweils als Index bei den Größen angegeben ist. Zeitliche Abhängigkeiten stehen in Klammern hinter den Größen, Zeitintervalle in eckigen Klammern, Zeitpunkte in runden Klammern. Sofern die Berechnung der jeweiligen Größe in diesem Abschnitt beschrieben ist, sind die betreffenden Gleichungsnummern angegeben.

Größe	Knoten A_j	Akquiseplan AP	Weg w^a	Kunden mit Abschluss	Kunden ohne Abschluss
Gewinn	Gewinn im Zeitpunkt t : $g_j(t)$; Gewinn im Zeitraum $[0, t]$: $g_j[0, t]$ (Aggregation über alle w , die durch A_j führen)	Gewinn im Zeitpunkt t : $g_{AP}(t)$; Gewinn im Zeitraum $[0, t]$: $g_{AP}[0, t]$ (3.40)	Gewinn im Zeitpunkt t : $g_w(t)$; Gewinn im Zeitraum $[0, t]$: $g_w[0, t]$ (nur w , die in A_N enden)	siehe Spalte <i>Knoten A_j</i> für A_N	Kunden ohne Abschluss erzeugen keinen Gewinn
Rendite	Rendite im Zeitpunkt t : $r_j(t)$; Rendite im Zeitraum $[0, t]$: $r_j[0, t]$ (Aggregation über alle w , die durch A_j führen)	Rendite im Zeitpunkt t : $r_{AP}(t)$; Rendite im Zeitraum $[0, t]$: $r_{AP}[0, t]$ (3.41)	Rendite im Zeitpunkt t : $r_w(t)$; Rendite im Zeitraum $[0, t]$: $r_w[0, t]$ (nur w , die in A_N enden)	siehe Spalte <i>Knoten A_j</i> für A_N	Kunden ohne Abschluss erzeugen keine Rendite

Tabelle 3.1: Übersicht von ökonomischen Größen, die in einer Simulation des Akquise-Prozesses berechnet werden könnten. Alle Größen sind Erwartungswerte für einen Kunden. Die Tabellenzeilen beziehen sich auf die verschiedenen Größen, die Tabellenspalten auf den betrachteten Aspekt der Akquise, der jeweils als Index bei den Größen angegeben ist. Zeitliche Abhängigkeiten stehen in Klammern hinter den Größen, Zeitintervalle in eckigen Klammern, Zeitpunkte in runden Klammern. Sofern die Berechnung der jeweiligen Größe in diesem Abschnitt beschrieben ist, sind die betreffenden Gleichungsnummern angegeben.

a. w bezeichnet hier einen bestimmten Weg von der Quelle bis zu einer der Senken, der durch seine Knotenfolge festzulegen ist.

Definition 3.14: *Ertragsfunktion.*

Die Abbildung $e : \underline{A} \rightarrow \mathbb{R}$ des Akquiseplans AP weist jeder Aktion $A_i \in \underline{A}$, $i \neq N$ die (erwarteten) Kosten und der Aktion $A_N \in \underline{A}$ die (erwarteten) Erträge zu, die bei Anwendung der Aktion auf *einen* Kunden entstehen, wobei

$$\begin{aligned} \forall A_i \notin \{A_{N-1}, A_N\} : e(A_i) &< 0, \\ e(A_N) &> 0, \\ e(A_{N-1}) &= 0. \end{aligned}$$

Dann definiert e eine *Ertragsfunktion*. □

Ausgehend von der Berechnung der Präsenzwahrscheinlichkeiten gemäß Satz 3.12 können bei gegebener Ertragsfunktion zahlreiche für die praktische Akquise interessante ökonomische Größen und ihr zeitlicher Verlauf berechnet werden. Eine für den Akquisiteur aufschlussreiche Information (die in Tabelle 3.1 nicht enthalten ist) ist z. B. der zu erwartende Ergebnisbeitrag eines Kunden im Knoten A_i . Je nach Wert dieser Größe könnte der Akquisiteur entscheiden, ob diese Aktion angewandt werden soll oder ob Kunden, auf die sie angewandt wurde, weiter beworben werden sollen. Weil der Schwerpunkt dieser Arbeit auf den Optimierungsproblemen in den Kapiteln 4 und 5 liegt, kann die Simulation des Akquise-Prozesses und seiner ökonomischen Größen hier nicht vertieft werden. In den folgenden Unterabschnitten werden einige Berechnungen exemplarisch angerissen und im Abschnitt 3.3 die Simulation kurz vorgestellt.

3.2.3.1 Ökonomische Größen für einzelne Knoten

Für einen Akquiseplan AP mit dem Wahrscheinlichkeitsraum (M, E, W) können mit den Ergebnissen aus Abschnitt 3.2.2 zu jedem Zeitpunkt $t = 0, 1, \dots$ der Vektor der Präsenzwahrscheinlichkeiten

$$p(t) = (p_1(t), \dots, p_N(t))$$

und die Übergangsmatrix

$$P_\omega(t+1)$$

angegeben werden. Mit ihrer Hilfe können für eine gegebene Ertragsfunktion gemäß Def. 3.14 die Erwartungswerte der Kosten, des Ertrags, des Gewinns und der Rendite berechnet werden. Die folgenden Berechnungen beziehen sich auf einen einzelnen Kunden. Die Bezeichnungen der ökonomischen Größen sind Tabelle 3.1 entnommen.

Berechnung zeitlich kumulierter Kosten¹³⁾ für die Knoten A_1, \dots, A_{N-2}

Kosten entstehen bei Anwendung einer Aktion. Die erste Aktion des Akquiseplans, A_1 wird mit Wahrscheinlichkeit eins zum Beginn der Akquise angewandt: $p_1(0) = 1$. Kosten entstehen somit nur für $t = 0$ und ändern sich im Verlauf der Akquise nicht mehr:

$$k_1[0, t] := k_1(0) = k_1 \quad (3.33)$$

für alle $t = 0, 1, \dots$. Für alle anderen Aktionen A_j , $j = 2, \dots, N-2$ entstehen Kosten immer dann, wenn Kunden in den betreffenden Knoten wechseln. Ihr Erwartungswert zum Zeitpunkt $t+1$ beträgt deshalb:

$$k_j[0, t+1] := k_j[0, t] + k_j \cdot \sum_{i=1}^{j-1} p_i(t) \cdot \psi_{ij}^{t+1} \quad \text{für } t = 0, 1, \dots \quad (3.34)$$

Berechnung des Ertrags für den Knoten A_N

Beim Eintreffen eines Kunden im Endknoten A_N entsteht ein Ertrag. Zur Zeit t befindet sich ein Kunde mit der Wahrscheinlichkeit $p_N(t)$ im Knoten A_N . Der Ertrag beträgt somit:

$$e_N[0, t] := e_N \cdot p_N(t) \quad \text{für } t = 0, 1, \dots \quad (3.35)$$

3.2.3.2 Ökonomische Größen für den ganzen Akquiseplan

Die erwarteten zeitlich kumulierten Kosten¹³⁾ für den ganzen Akquiseplan ergeben sich durch Addition der entsprechenden Kosten über alle Knoten A_1, \dots, A_{N-2} :

$$k_{AP}[0, t+1] := \sum_{j=1}^{N-2} k_j[0, t+1] \quad \text{für } t = 0, 1, \dots \quad (3.36)$$

Nach Einsetzen von (3.34) und Umformung ergibt sich:

$$k_{AP}[0, t+1] = k_{AP}[0, t] + \sum_{j=1}^{N-2} \left[k_j \cdot \sum_{i=1}^{j-1} [p_i(t) \cdot \psi_{ij}^{t+1}] \right] \quad \text{für } t = 0, 1, \dots \quad (3.37)$$

Erträge entstehen nur mit Aktion A_N . Der erwartete Ertrag des Akquiseplans ist:

$$e_{AP}[0, t] := e_N[0, t] \quad \text{für } t = 0, 1, \dots, \quad (3.38)$$

13. Der Einfachheit halber werden Verzinsungseffekte nicht berücksichtigt

also mit (3.35):

$$e_{AP}[0, t] := e_N \cdot p_N(t) \text{ für } t = 0, 1, \dots \quad (3.39)$$

Gewinn und Rendite sind ökonomische Größen des ganzen Akquiseplans, da sie die Betrachtung aller Aktionen erfordern. Ihr zeitlicher Verlauf ergibt sich unmittelbar aus (3.33) – (3.35):

$$g_{AP}[0, t] := e_{AP}[0, t] - k_{AP}[0, t] \text{ für } t = 0, 1, \dots \quad (3.40)$$

sowie

$$r_{AP}[0, t] := \frac{g_{AP}[0, t]}{k_{AP}[0, t]} \text{ für } t = 0, 1, \dots \quad (3.41)$$

Alle Größen sind Erwartungswerte für einen Kunden. Interessiert der Wert einer Größe für eine Menge von Kunden, so ist der entsprechende Wert mit der Anzahl der Kunden zu multiplizieren.

3.3 Simulation der Kundenakquisition mit Akquiseplänen

In diesem Abschnitt wird eine Simulation der ökonomischen Größen aus Abschnitt 3.2.3 auf Grundlage des in den Abschnitten 3.1 und 3.2 eingeführten Modells vorgestellt. Diese Simulation eines Akquise-Prozesses könnte z. B. dann interessant sein, wenn das Budget einer Marketing-Kampagne begrenzt ist und vorab geklärt werden soll, wieviele Kunden unter Einhaltung des Budgets beworben werden können.

Die Berechnung des zeitlichen Verlaufs der Erwartungswerte der ökonomischen Größen ist nur eine Möglichkeit der Simulation. Eine andere wäre z. B. die Monte-Carlo-Simulation des Weges der einzelnen Kunden durch den Akquiseplan. Weil der Schwerpunkt dieser Arbeit nicht auf der Simulation liegt, wird diese Möglichkeit hier nicht untersucht. Sie wird aber im Abschnitt 4.3 zur Lösung des stochastischen Problems der optimalen Beschickung von Akquiseplänen eingesetzt.

3.3.1 Bereitzustellende Informationen

Die Berechnung von Kosten, Ertrag, Gewinn und Rendite erfolgt für den ganzen Akquiseplan. Dazu sind folgende Informationen bereitzustellen:

- die Struktur des *Akquiseplans* AP , das heißt Festlegung von Knoten (Aktionen) und Kanten (Reaktionen),
- die zeitinvarianten *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten*, p_{jk} ,
- die zeitinvarianten *zeitlichen Punktwahrscheinlichkeiten des Knotenwechsels*, $\varphi_{ij}(\tau)$,
- Festlegung der *Ertragsfunktion* e durch Angabe der erwarteten Kosten der Aktionen A_1, \dots, A_{N-2} sowie des erwarteten Ertrags der Aktion A_N .

Für die Simulation des Akquiseplans in Abb. 3.5 wurden diese Größen von einem Experten geschätzt. Ihre Werte des hier betrachteten Fallbeispiels können seiner Implementierung entnommen werden, die im nächsten Abschnitt beschrieben wird.

3.3.2 Implementierung des Verfahrens

Die iterative Berechnung der Präsenzwahrscheinlichkeiten $p_i(t)$ für zeitinvariante p_{ij}^t und π_{ij}^t wird gemäß Satz 3.12 von dem Algorithmus Alg. 3.1 umgesetzt. Eine um die Berechnung der ökonomischen Größen aus Abschnitt 3.2.3 erweiterte Variante dieses Algorithmus wurde mit *Maple*¹⁴⁾ implementiert. In diesem Abschnitt wird der Programm-Kode erläutert. Er befindet sich in der *Maple*-Datei *simulation.mws*, die von der Homepage des Verfassers¹⁵⁾ herunter geladen werden kann. Er ist prozedural aufgebaut und in drei Hauptabschnitte unterteilt, die im Programm-Kode entsprechend kommentiert sind.

Struktur des Akquiseplans und Eingabedaten

Der Programmabschnitt, der die Struktur des Akquiseplans festlegt und die Eingabedaten enthält, heißt in *simulation.mws* *Definition of graph structure and initialization of transition probabilities, time density functions, and probabilities of presence*. Der Vektor der *Präsenzwahrscheinlichkeiten* zum Akquise-Beginn, $p(0)$ (vgl. Def. 3.6) wird von der Funktion

```
initd := proc(N::posint)
```

initialisiert. Die zeitinvarianten *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten*, p_{ij} , $i, j \in \{1, \dots, N\}$ (vgl. Def. 3.8) sind im Programm-Kode „hart“ kodiert¹⁶⁾ und werden von der Funktion

```
initp := proc(N::posint)
```

initialisiert. Sie legen auch die Struktur des Akquiseplans fest, so dass keine Adjazenzmatrix angegeben werden muss. Die zeitinvarianten *Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten*, π_{ij} , $i, j \in \{1, \dots, N\}$ (vgl. Def. 3.10) sind ebenfalls im Programm-Kode „hart“ kodiert¹⁶⁾ und werden von der Funktion

```
initPhi := proc(N::posint)
```

initialisiert. Der Parameter N ist bei allen Funktionen gleich der Anzahl der Knoten des Akquiseplans.

14. *Maple* ist ein an der Waterloo-Universität (Kanada) entwickeltes Computeralgebrasystem, das von der Firma *Waterloo Maple Software* (<http://www.maplesoft.com>) vertrieben wird. Für die vorliegende Arbeit wurde Version V, Release 5 verwendet.

15. <http://vieta.math.tu-cottbus.de/~klweber>

16. Bei der für das Fallbeispiel im Abschnitt 4.4 implementierten Variante werden diese Daten aus Textdateien eingelesen.

```

// Initialisierung der Parameter
Für (i = 1, ..., (N-2)) führe aus
    Für (j = (i+1), ..., N) führe aus {
        Eingabe  $p_{ij}$ ;           //  $p_{ij} \in [0, 1], \sum_{j=i+1}^N p_{ij} = 1$ 
        Eingabe  $\varphi_{ij}$ ;           //  $\sum_{\tau=0}^{\infty} \varphi_{ij}(\tau) = 1$ 
    }
// Initialisierung des Abbruchkriteriums (Endzeit)
Eingabe  $t_{\max}$ ;
// Initialisierung von Hilfsvariablen
Für (i = 1, ..., (N-2)) führe aus  $\omega_i^0(0) = 1$ ;
// Initialisierung von Anfangszustand
Setze  $(p(0) = (p_1(0), \dots, p_N(0))) := (1, 0, \dots, 0)$ ;
// Hauptprogramm
Für (t = 0, ..., ( $t_{\max} - 1$ )) führe aus {
    Für (i = 2, ..., N) führe aus
        für (j = 1, ..., (i-1)) setze  $\psi_{ij}^{t+1} := 0$ ;

    Setze  $\psi_{N-1, N-1}^{t+1} = \psi_{N, N}^{t+1} := 1$ ;
    Setze  $\psi_{N-1, N}^{t+1} := 0$ ;
    Für (i = 1, ..., (N-2)) führe aus {
        Für (j = (i+1), ..., N) setze  $\pi_{ij}^t(t) := \frac{p_{ij}\varphi_{ij}(t)}{\sum_{k=i+1}^N p_{ik} \sum_{\tau=t}^{\infty} \varphi_{ik}(\tau)}$ ;

        Setze  $\pi_{ii}^t(t) := 1 - \sum_{j=i+1}^N \pi_{ij}^t(t)$ ;
        Wenn (t > 0) dann führe aus {
            Für ( $\tau = 1, \dots, t$ ) setze  $\omega_i^t(\tau) := \begin{cases} \frac{p_i(t-1)\omega_i^{t-1}(\tau-1)\pi_{ii}^{t-1}(\tau-1)}{p_i(t)} & \text{if } p_i(t) > 0 \\ 0 & \text{sonst} \end{cases}$ ;

            Setze  $\omega_i^t(0) := 1 - \sum_{\tau=1}^t \omega_i^t(\tau)$ ;
        }

        Für (j = (i+1), ..., N) setze  $\psi_{ij}^{t+1} := \sum_{\tau=0}^t \omega_i^t(\tau)\pi_{ij}^t(\tau)$ ;

        Setze  $\psi_{ii}^{t+1} := 1 - \sum_{j=i+1}^N \psi_{ij}^{t+1}$ ;
    }

     $P_{\omega}(t+1) := (\psi_{ij}^{t+1})_{i,j=1, \dots, N}$ ;
     $p(t+1) := (p_1(t+1), \dots, p_N(t+1)) = p(t) \cdot P_{\omega}(t+1)$ ;
    Ausgabe:  $p(t+1)$ ;
}

```

Algorithmus 3.1: Algorithmus zur iterativen Berechnung der Präsenzwahrscheinlichkeiten $p(t)$, $t = 0, \dots, t_{\max} - 1$ gemäß Satz 3.12.

Algorithmen und Datenausgabe

Dieser in `simulation.mws` *Algorithms* überschriebene Programmabschnitt enthält als Hauptprogramm

```
main := proc(N::posint, tmax::posint, epsilon::numeric,
  cost::array(), d::array(), p::array(), Phi::array(), dis,
  remain, pos, neg, co)
```

im Wesentlichen den Algorithmus Alg. 3.1, erweitert um die Berechnung der in den Abschnitten 3.2.3.1 und 3.2.3.2 eingeführten ökonomischen Größen. Die Funktionsparameter haben folgende Bedeutung:

- `N`: Anzahl der Knoten des Akquiseplans;
- `tmax`: maximale Anzahl der Zeitschritte, Abbruchkriterium der Simulation¹⁷⁾;
- `epsilon`: Abbruchkriterium¹⁷⁾ der Simulation;
- `cost`: Vektor mit den Werten der *Ertragsfunktion* $e(A_i)$, $i = 1, \dots, N$ (vgl. Def. 3.14);
- `d`: Vektor der *Präsenzwahrscheinlichkeiten* zum Akquise-Beginn, $p(0)$ (vgl. Def. 3.6)¹⁸⁾;
- `p`: Matrix der zeitinvarianten *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten*, p_{ij} , $i, j \in \{1, \dots, N\}$ (vgl. Def. 3.8);
- `Phi`: Matrix mit den *zeitlichen Verteilungsfunktionen des Knotenwechsels*, $\phi_{ij}(t)$, $t \in \mathbb{N}_0$ (vgl. Def. 3.9);
- `dis`: Folge der *Präsenzwahrscheinlichkeiten*, $(p(t))_{t \geq 0}$ (vgl. Def. 3.6);
- `remain`: Folge der Summen der *Präsenzwahrscheinlichkeiten* in den Knoten A_1, \dots, A_{N-2} : $(\sum_{i=1}^{N-2} p_i(t))_{t \geq 0}$, zur Bestimmung des Abbruchs der Simulation¹⁷⁾;
- `pos`: Folge der *Präsenzwahrscheinlichkeit im Knoten N*: $(p_N(t))_{t \geq 0}$;
- `neg`: Folge der *Präsenzwahrscheinlichkeit im Knoten N - 1*: $(p_{N-1}(t))_{t \geq 0}$;
- `co`: Matrix mit den Erwartungswerten der Kosten und Erträge im Zeitraum $[0, t]$: $k_j[0, t]$ für $j = 1, \dots, N-2$ bzw. $e_N[0, t]$ (vgl. (3.33) – (3.35) und Tab. 3.1).

Das Programm `main` gibt den Zeitschritt an, bei dem eines der Abbruchkriterien¹⁷⁾ erfüllt ist.

Weiterhin enthält der Programmabschnitt *Algorithms* Funktionen zur Ausgabe und Darstellung sowohl der Eingabedaten als auch der berechneten Größen. Die Aufrufparameter aller nachfol-

17. Die Simulation wird abgebrochen, wenn $\sum_{i=1}^{N-2} p_i(t) < \epsilon$ oder $t \geq t_{\max}$.

18. Die Programmvariable kann nicht mit p bezeichnet werden, weil p für die *Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten* verwendet wird.

genden Funktionen sind oben für die Funktion `main` erklärt und werden hier nicht wiederholt. Die zeitlichen Verteilungsfunktionen des Knotenwechsels, $\phi_{ij}(t)$, $t \in \mathbb{N}_0$ werden von der Funktion

```
cdfplot := proc(N::posint, p::array(), Phi::array())
```

graphisch ausgegeben. Ein Beispiel zeigt Abb. 3.6. Den zeitlichen Verlauf der *Präsenzwahr-*

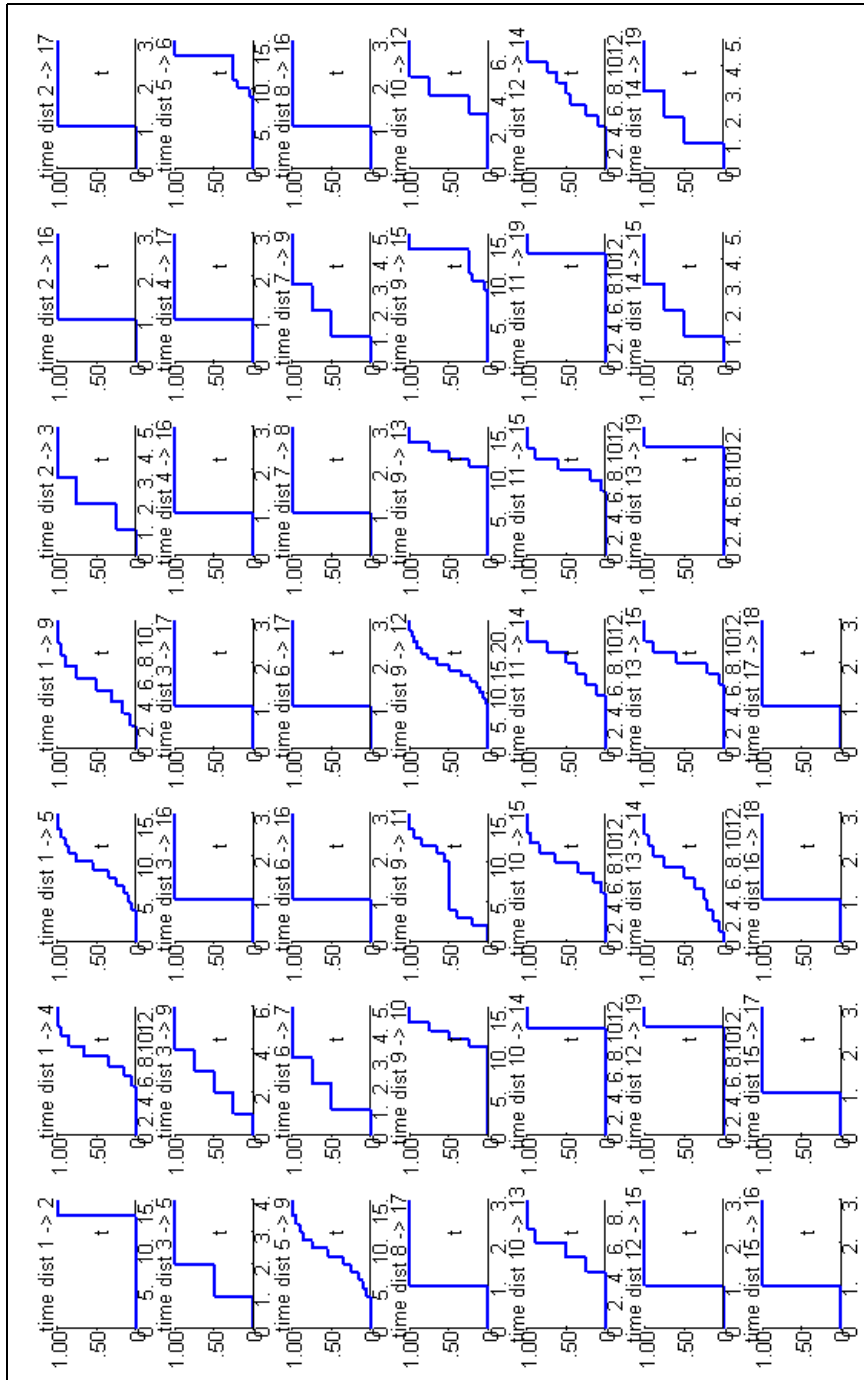


Abbildung 3.6: Zeitliche Verteilungsfunktionen des Knotenwechsels $\phi_{ij}(t)$, $t \in \mathbb{N}_0$ für i, j mit $p_{ij} > 0$; die Zeitskalen sind dem Verlauf der Funktionen angepasst.

scheinlichkeiten in den beiden Endknoten, $p_{N-1}(t)$, $p_N(t)$, $t = 0, 1, \dots$ sowie der Summe dieser Wahrscheinlichkeiten zeichnet die Funktion

```
convplot := proc(N::posint, tmax::posint, remain, pos, neg).
```


Ein Beispiel zeigt Abb. 3.7. Den zeitlichen Verlauf der *Präsenzwahrscheinlichkeiten* für alle Knoten, $p_i(t)$, $t = 0, 1, \dots$ gibt als graphische Animation die Funktion

```
displot := proc(N::posint, tmax::posint, d)
```

aus und in numerischer Darstellung die Funktion

```
disprint := proc(N::posint, tmax::posint, d, remain).
```

Außer dem Vektor der *Präsenzwahrscheinlichkeiten* $p(t)$ gibt diese Funktion für jeden Zeitschritt t auch die Summe der Präsenzwahrscheinlichkeiten in den Endknoten aus. Den Verlauf der Kosten und Erträge im Zeitraum $[0, t]$: $k_j[0, t]$ für $j = 1, \dots, N-2$ bzw. $e_N[0, t]$ ¹⁹⁾ gibt die Funktion

```
costplot := proc(N::posint, tmax::posint, co)
```

als graphische Animation aus. Den zeitlichen Verlauf des Gewinns im Zeitraum $[0, t]$: $g_{AP}[0, t]$ (vgl. (3.40) und Tab. 3.1) gibt die Funktion

```
profitplot := proc(N::posint, tmax::posint, co)
```

aus, für die Abb. 3.8 ein Beispiel zeigt.

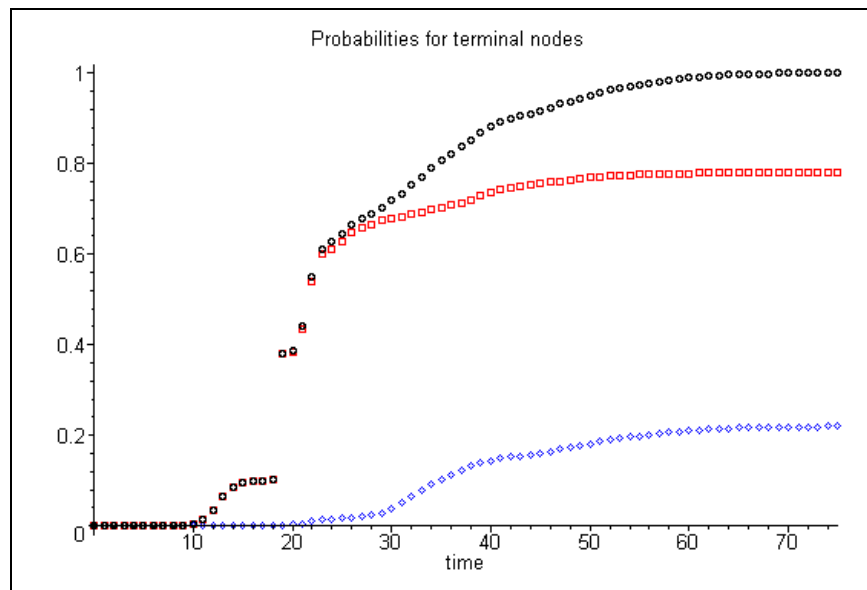


Abbildung 3.7: Zeitliche Entwicklung der *Präsenzwahrscheinlichkeiten* $p_{N-1}(t)$ (rot) und $p_N(t)$ (blau) sowie ihrer Summe (schwarz) für den Akquiseplan in Abb. 3.5. Auf der Abszisse ist die Zeit t in Tagen abgetragen, auf der Ordinate die Wahrscheinlichkeit.

19. Die Aktion A_{N-1} verursacht weder Kosten noch erzeugt sie Ertrag (siehe Def. 3.14). In der Ausgabe der Funktion `costplot` wird ihr für alle $t \in \mathbb{N}_0$ der Wert Null zugeordnet.

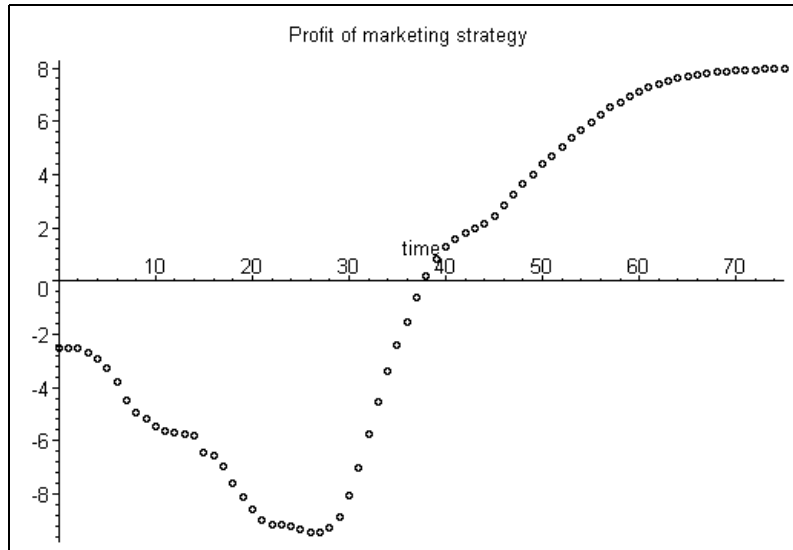


Abbildung 3.8: Zeitliche Entwicklung des Gewinns $g_{AP}[0, t]$ für den Akquiseplan in Abb. 3.5. Auf der Abszisse ist die Zeit t in Tagen abgetragen, auf der Ordinate der erwartete Gewinn für *einen* Kunden.

Simulationsprogramm

Dieser Programmabschnitt ist in `simulation.mws` mit *Algorithms* bezeichnet. Den Programm-Kode zeigt Abb. 3.9. Die einzelnen Schritte haben folgende Bedeutung: Nach dem Setzen der Knotenanzahl N und der maximalen Anzahl an Zeitschritten t_{\max} werden die Eingabegrößen initialisiert:

- Die *Ertragsfunktion* `cost` wird explizit durch Nennung aller Werte für $e(A_i)$, $i = 1, \dots, N$ angegeben.
- Die Präsenzwahrscheinlichkeiten $p(0)$, die Wahrscheinlichkeiten p_{ij} , $i, j \in \{1, \dots, N\}$ und die zeitlichen Verteilungsfunktionen $\phi_{ij}(t)$, $t \in \mathbb{N}_0$ werden von den Funktionen `initd`, `initp` und `initPhi` initialisiert. Die Werte sind im Quellcode der Funktionen „hart“ kodiert.
- Danach gibt die Funktion `cdfplot` die zeitlichen Verteilungsfunktionen aus.
- Die Funktion `main` berechnet iterativ die Präsenzwahrscheinlichkeiten $p_i(t)$, die Übergangsmatrizen $P_{\omega}(t+1)$, solange bis entweder die angegebene Anzahl t_{\max} an Zeitschritten erreicht ist oder für die Summe der Präsenzwahrscheinlichkeiten $\sum_{i=1}^{N-2} p_i(t) < \varepsilon$ gilt. Im gezeigten Beispiel ist $\varepsilon = 0,01$. Die Anzahl der Zeitschritte bis zum Abbruch der Berechnung ist der Rückgabewert `tend`.
- Danach werden die Berechnungsergebnisse mit Hilfe der Funktionen `disprint`, `displot`, `convplot`, `costplot`, `profitplot` ausgegeben bzw. dargestellt.

```

N := 19:
tmax := 100:
cost := array(1..N, [-2.50, -1.00, -2.50, -1.00, -6.00, -1.00, -2.50,
-1.00, -10.00, -15.00, -5.00, -15.00, -25.00, -2.50, -1.00, 0.00,
0.00, 0.00, 100.00 ]):
d := initd(N):
p := initp(N):
Phi := initPhi(19):
cdfplot(N, p, Phi, false);
tend := main(N, tmax, 0.01, cost, d, p, Phi, dis, remain, pos, neg, co):
disprint(N, tend, dis, remain);
displot(N, tend, dis, true);
convplot(N, tend, remain, pos, neg);
costplot(N, tend, co, true);
profitplot(N, tend, co);

```

Abbildung 3.9: *Maple*-Hauptprogramm in simulation.mws.

3.3.3 Durchführung und Auswertung der Simulation – Fallbeispiele

Von einem Experten mit Erfahrung im Vertrieb von Versicherungen wurde der Akquiseplan in Abb. 3.5 aufgestellt und die für die Simulation nötigen Eingabedaten (siehe Abschnitt 3.3.1) festgelegt. Sie sind im Programm-Kode des *Maple*-Programms simulation.mws „hart“ kodiert (siehe Abschnitt 3.3.2). Die zeitliche Verteilungsfunktion des Knotenwechsels ist auch in Abb. 3.6 dargestellt. Zur Illustration zeigt Tab. 3.2 die Werte der Ertragsfunktion für alle Aktionen und vier Fallbeispiele. Die Fallbeispiele sind wie folgt definiert:

Aktion	Wert der Ertragsfunktion (Def. 3.14) ^a			
	Fallbeispiel 1 „Normalfall“	Fallbeispiel 2 „doppelte Kosten, glei- cher Ertrag“	Fallbeispiel 3 „gleiche Kos- ten, halber Ertrag“	Fallbeispiel 4 „doppelte Kosten, dop- pelter Ertrag“
A ₁ : unpersönlicher Mas- senbrief	-2,50 €	-5,00 €	-2,50 €	-5,00 €
A ₂ : Auslese 1	-1,00 €	-2,00 €	-1,00 €	-2,00 €
A ₃ : Telefonat 1	-2,50 €	-5,00 €	-2,50 €	-5,00 €
A ₄ : Auslese 2	-1,00 €	-2,00 €	-1,00 €	-2,00 €
A ₅ : persönlicher Brief mit detaillierten Produktinfor- mationen	-6,00 €	-12,00 €	-6,00 €	-12,00 €
A ₆ : Auslese 3	-1,00 €	-2,00 €	-1,00 €	-2,00 €

Tabelle 3.2: Werte der Ertragsfunktion (Kosten/Erträge) für den Akquiseplan in Abb. 3.5 für vier Fallbeispiele.

Aktion	Wert der Ertragsfunktion (Def. 3.14) ^a			
	Fallbeispiel 1 „Normalfall“	Fallbeispiel 2 „doppelte Kosten, gleicher Ertrag“	Fallbeispiel 3 „gleiche Kosten, halber Ertrag“	Fallbeispiel 4 „doppelte Kosten, doppelter Ertrag“
A ₇ : Telefonat 2	-2,50 €	-5,00 €	-2,50 €	-5,00 €
A ₈ : Auslese 4	-1,00 €	-2,00 €	-1,00 €	-2,00 €
A ₉ : persönlicher Brief mit Angebot	-10,00 €	-20,00 €	-10,00 €	-20,00 €
A ₁₀ : persönlicher Brief mit verbessertem Angebot	-15,00 €	-30,00 €	-15,00 €	-30,00 €
A ₁₁ : Vertrag zusenden	-5,00 €	-10,00 €	-5,00 €	-10,00 €
A ₁₂ : Übergabe des Vertrags in der Geschäftsstelle	-15,00 €	-30,00 €	-15,00 €	-30,00 €
A ₁₃ : Übergabe des Vertrags beim Kunden	-25,00 €	-50,00 €	-25,00 €	-50,00 €
A ₁₄ : Telefonat 3	-2,50 €	-5,00 €	-2,50 €	-5,00 €
A ₁₅ : Auslese 5	-1,00 €	-2,00 €	-1,00 €	-2,00 €
A ₁₆ : Reißwolf (Vernichtung des Kundendatensatzes)	0,00	0,00	0,00	0,00
A ₁₇ : Ablage: zu den Akten (Zurücklegen des Kundendatensatzes für künftige Akquisen)	0,00 €	0,00 €	0,00 €	0,00 €
A ₁₈ : kein Vertrag (negatives Akquise-Ergebnis)	0,00	0,00	0,00	0,00
A ₁₉ : gültiger Vertrag (positives Akquise-Ergebnis)	100,00 €	100,00 €	50,00 €	200,00 €

Tabelle 3.2: Werte der Ertragsfunktion (Kosten/Erträge) für den Akquiseplan in Abb. 3.5 für vier Fallbeispiele.

a. Die Werte der Ertragsfunktion stammen nicht von einem Experten.

Fallbeispiel 1: „Normalfall“, auf den die anderen Fallbeispiele bezogen werden.

Fallbeispiel 2: Gleicher Ertrag wie im Normalfall, aber doppelte Kosten. Dieser Fall könnte z. B. eine Kostenexplosion bei gleichbleibendem Ertrag (mangels Kaufkraftzuwachs) darstellen.

Fallbeispiel 3: Gleiche Kosten wie im Normalfall, aber nur halber Ertrag. Dieser Fall könnte z. B. einen Ertragseinbruch durch Billigangebote oder starke Rabattierung ohne entsprechende Kostensenkung darstellen.

Fallbeispiel 4: Doppelte Kosten als im Normalfall und auch doppelter Ertrag. Dieser Fall könnte z. B. verstärkte Vertriebsanstrengungen (und deshalb erhöhte Kosten) bei damit einhergehenden höheren Erträgen (durch den Verkauf höherwertiger Produkte) darstellen.

Die Ergebnisse der Simulation für Fallbeispiel 1 sind in den Abbildungen 3.7 und 3.8 dargestellt. Mit ihrer Hilfe können die folgenden Fragen beantwortet werden:

- Wann erreicht die Akquise den Kostendeckungspunkt („break-even point“)?
- Sollte die Akquise nach einer bestimmten Zeit abgebrochen werden, weil der Gewinn nicht weiter zunimmt?

Abb. 3.7 zeigt die zeitliche Entwicklung der Präsenzwahrscheinlichkeiten für die beiden Endknoten, positiver und negativer Vertragsabschluss. Entsprechend der Wahrscheinlichkeiten können die ersten negativen Akquise-Abschlüsse etwa ab dem 10. Tag nach Akquisebeginn erwartet werden. Ab dem 18. Tag ist ihre sprunghafte Zunahme zu erwarten. Positive Akquise-Abschlüsse sind in nennenswertem Umfang erst ab dem 26. Tag zu erwarten. Etwa ab dem 40. Tag nach Akquisebeginn werden sich durchschnittlich über 90 % aller Kunden positiv oder negativ entschieden haben.

Abb. 3.8 zeigt die erwartete zeitliche Entwicklung des Gewinns. Weil entsprechend der zeitlichen Entwicklung der Präsenzwahrscheinlichkeiten in Abb. 3.7 positive Akquiseabschlüsse erst deutlich später als negative Abschlüsse zu erwarten sind, kann der Zeitpunkt, an dem sich Kosten und Erträge die Waage halten („break-even point“), erst ab dem 32. Tag erwartet werden. Ab dem 40. Tag steigt der Gewinn nur noch langsam und nur um einen kleinen Betrag. Daher könnten alle Aktionen, die nicht im Zusammenhang mit einem Vertragsabschluss stehen, in dieser Phase der Akquise eingestellt werden.

Der Vergleich aller Fallbeispiele in Abb. 3.10 zeigt folgende Ergebnisse:

- **Fall 1:** Trotz des hohen Ertrags pro Vertragsabschluss (100 €) und einer eher hohen Abschlussrate von etwa 40 % (siehe Abb. 3.7) ergibt sich pro Kunde nur ein erwarteter Gewinn von etwa 8 €.
- **Fall 2:** Bei einer Verdoppelung der Kosten wird kein Gewinn mehr gemacht.
- **Fall 3:** Bei einer Halbierung des durchschnittlichen Ertrags pro Vertragsabschluss wird ebenfalls keinen Gewinn mehr gemacht.
- **Fall 4:** Bei Verdoppelung der Kosten und gleichzeitiger Verdoppelung des durchschnittlichen Ertrags pro Vertragsabschluss bleibt der Gewinn – wie erwartet werden könnte – nicht auf dem Niveau von Fall 1, sondern steigt von 8 € auf etwa 16 € an. Der Gewinn $g_{AP}[0, t]$ ist offensichtlich linear von der Ertragsfunktion abhängig.

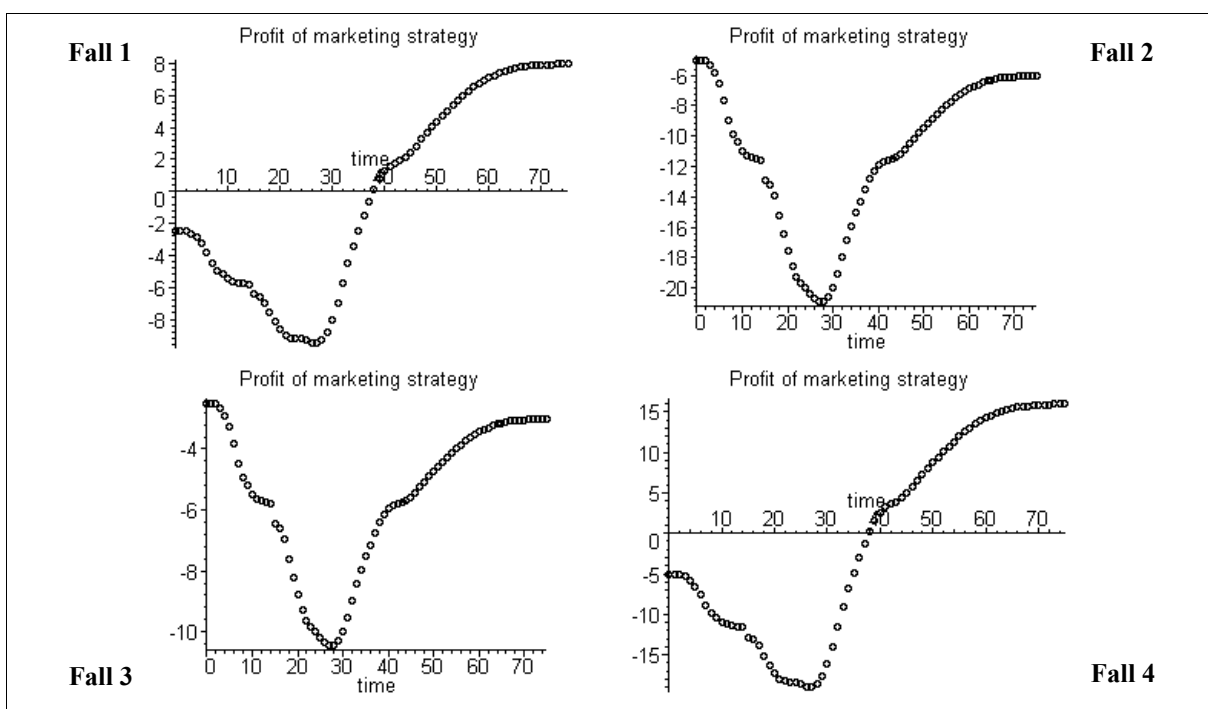


Abbildung 3.10: Vergleich der Fallbeispiele 1 – 4 (siehe Tab. 3.2): Zeitliche Entwicklung des Gewinns $g_{AP}[0, t]$ für den Akquiseplan in Abb. 3.5. Auf der Abszisse ist die Zeit t in Tagen abgetragen, auf der Ordinate der erwartete Gewinn für *einen* Kunden.

3.4 Unscharfe stochastische Optimierung mit Akquiseplänen

Das mathematische Modell in Abschnitt 3.2 zeigt im Satz 3.12 die iterative Berechnung der Wahrscheinlichkeitsverteilung für Ereignisse $A(i, t)$, „ein Kunde befindet sich zum Zeitpunkt t im Knoten A_i “ (vgl. Def. 3.5). Die darauf aufbauende Simulation in Abschnitt 3.3 erlaubt die Berechnung der Wahrscheinlichkeiten, z. B. für die beiden Endknoten, wodurch Rückschlüsse auf den Verlauf der Akquise möglich sind. Weiterhin können mit der Simulation der zeitliche Verlauf von Erwartungswerten ökonomischer Kenngrößen, wie Kosten, Erträge, Gewinn und Rendite, berechnet werden. Diese Information kann die Planung von Akquisen unterstützen.

Sowohl Modellierung als auch Simulation beruhen auf der Annahme, dass alle Aktionen „sofort“, das heißt bis zum nächsten diskreten Zeitpunkt durchgeführt werden können. Im Allgemeinen wird diese Annahme nicht zutreffen. Insbesondere die Durchführung von Akquisekampagnen mit großen Kundenmengen verlangt die Berücksichtigung von Kapazitätsschranken für die Durchführung der Akquiseaktionen eines Akquiseplans. Solche Kapazitätsbeschränkungen hängen von mehreren Faktoren ab, z. B.

- die Anzahl der an der Akquiseaktion beteiligten Arbeitskräfte,
- der Anteil ihrer regulären Arbeitszeit für die Akquiseaktion,
- die Möglichkeit, die für die Akquise verfügbare Arbeitszeit durch Überstunden zu verlängern,
- und die Anzahl der Überstunden für jede Arbeitskraft,

- die Zuordnung der Arbeitskräfte zu den einzelnen Aktionen und ihre Austauschbarkeit.

Offensichtlich ist eine Modellierung der Kapazitätsbeschränkungen nicht trivial. Eine einfache Möglichkeit ist die Befragung eines Experten und die Angabe der Kapazitäten als unscharfe Schranken. Das sind linguistische Variablen (vgl. Def. 2.14) wie

- Anzahl der Briefe, die an einem Tag geschrieben werden können,
- Anzahl der Kunden-Telefonate, die an einem Tag geführt werden können,
- Anzahl der Kundenbesuche, die an einem Tag durchgeführt werden können.

Die Werte dieser linguistischen Variablen sind linguistische Terme (vgl. Def. 2.13) wie „mindestens 20“ (Briefe), „etwa 35“ (Telefonate), „höchstens 5“ (Kundenbesuche), die ihre Bedeutung durch unscharfe Mengen erhalten. Das sind unscharfe Mengen aus $\mathcal{IF}(\mathbb{N}_0)$, deren Zugehörigkeitsfunktionen folgende Eigenschaften besitzen [99]:

$$\mu(x) = 1 \text{ für } x \leq a \text{ und } \mu(x) = 0 \text{ für } x \geq b \quad (3.42)$$

sowie

$$\mu(x) \text{ monoton abnehmend für } a < x < b \quad (3.43)$$

für $a, b, x \in \mathbb{N}_0$. Ein Beispiel zeigt Abb. 3.11. Eine formale Definition wird in Kapitel 4 eingeführt.

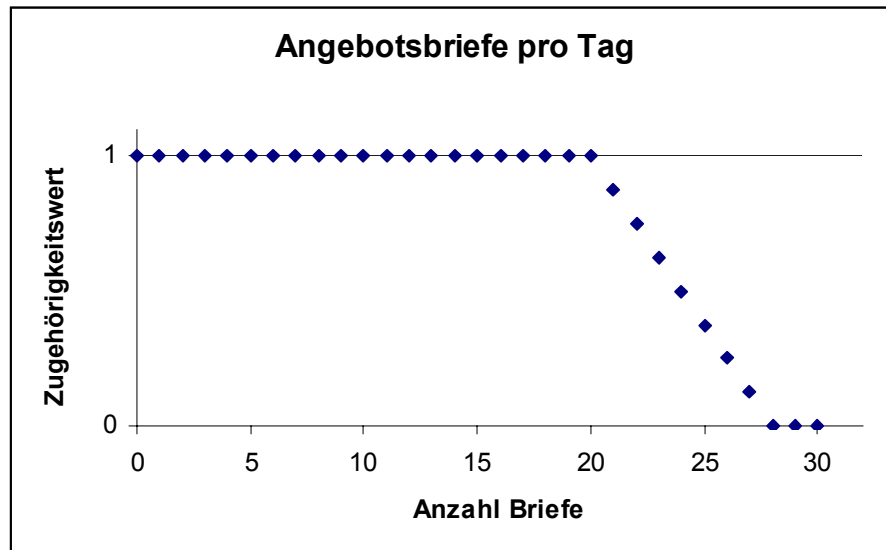


Abbildung 3.11: Beispiel einer unscharfen Kapazitätsbeschränkung.

Das Bestreben, möglichst viele Kunden unter Einhaltung der Kapazitätsschranken abzuarbeiten, ist ein Optimierungsproblem. Wie oben erwähnt, sind die Ereignisse $A(i, t)$ zufällig. Wenn mehrere Kunden gleichzeitig dem Akquiseplan zugeführt werden, ist die Anzahl der Kunden, die sich zu einem bestimmten Zeitpunkt in einem bestimmten Knoten befinden, eben-

falls eine Zufallsgröße. Das Optimierungsproblem ist somit stochastisch. Aufgrund der unscharf beschriebenen Kapazitätsschranken ist es zugleich unscharf. Tatsächlich können mehrere *stochastische unscharfe Optimierungsprobleme* beschrieben werden. Ihre mathematische Formulierung und Lösung wird in Kapitel 4 behandelt. Hier folgt zunächst ihre verbale Beschreibung.

Erstes Optimierungsproblem

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan gleichzeitig unter Einhaltung der Kapazitätsschranken der Akquiseaktionen zugeführt werden kann?

Zweites Optimierungsproblem

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan innerhalb eines gegebenen Zeitraums unter Einhaltung der Kapazitätsschranken für Akquiseaktionen zugeführt werden kann, und wie ist sie über den Zeitraum verteilt?

Drittes Optimierungsproblem

Wie muss eine vorgegebene Anzahl von Kunden einem Akquiseplan unter Einhaltung der Kapazitätsschranken zugeführt werden, damit der dafür benötigte Zeitraum möglichst kurz ist?

Beim zweiten Optimierungsproblem kann die Entscheidung über die Anzahl der dem Akquiseplan in einem Zeitschritt zuzuführenden Kunden entweder vor der Akquise getroffen werden oder vor dem jeweiligen Zeitschritt im Verlauf der Akquise, also dann, wenn die Realisierung des Kundenverhaltens im letzten Zeitschritt vorliegt. Bei der ersten Vorgehensweise ist das Optimierungsproblem vom Typ *jetzt-und-sofort* („here-and-now“), bei der zweiten Vorgehensweise ist es vom Typ *abwarten-und-Teetrinken* („wait-and-see“) [132, 168]²⁰.

Alle drei Optimierungsprobleme sind ganzzahlig. Im Kapitel 4 wird gezeigt, dass die drei stochastischen unscharfen Optimierungsprobleme aufeinander aufbauen. Für den Rest der Arbeit wird auf sie als *erstes Optimierungsproblem*, *zweites Optimierungsproblem* und *drittes Optimierungsproblem* Bezug genommen. Alle drei Optimierungsprobleme werden unter der Bezeichnung *maximale Beschickung von Akquiseplänen* zusammengefasst.

20. Für die in der englischen Fachliteratur gebräuchlichen Bezeichnungen „here-and-now“ und „wait-and-see“ konnten in der deutschen Fachliteratur keine deutschen Entsprechungen gefunden werden. Die hier gewählten Übersetzungen „jetzt-und-sofort“ und „abwarten-und-Teetrinken“ treffen den Sinn der englischen Bezeichnungen und werden nicht als „geschnörkelt“ empfunden.

4 Unscharfe stochastische Optimierung – Optimale Beschickung eines Akquiseplans

In diesem Kapitel werden die stochastischen unscharfen Optimierungsprobleme aus Abschnitt 3.4 mathematisch beschrieben und gezeigt, wie sie gelöst werden können. Dabei wird schrittweise vorgegangen. Zunächst wird unter der vereinfachenden Annahme eines deterministischen Kundenverhaltens im Abschnitt 4.1 die Behandlung unscharfer Kapazitätsschranken geklärt. Abschnitt 4.2 behandelt drei Optimierungsprobleme mit einem Erwartungswertansatz sowie dem Verfahren der *Optimierung mit zufälligen Nebenbedingungen* („chance constrained programming“). Da beide nur bedingt geeignet sind, untersucht Abschnitt 4.3 die Struktur des Optimierungsproblems und stellt es als Variante eines *Mehrgüterfluss-Problems* dar. Für seine Modellierung wird der Akquiseplan zum *zeitlichen Akquiseplan* erweitert. In diesem Abschnitt wird auch die Erzeugung des deterministischen Äquivalents des stochastischen Problems mit Hilfe von *Szenarien* gezeigt sowie die beiden darauf aufbauenden Lösungsverfahren: *Mittelwert-Verfahren* und erweitertes *Lagrange-Verfahren*. Deren Umsetzung in ein C++-Computerprogramm unter Einsatz von *CPLEX* wird im Abschnitt 4.4 genau beschrieben. Zur Bewertung der Verfahren wird mit einem wirklichkeitsnahen *Fallbeispiel* ein umfangreiches Untersuchungsprogramm durchgeführt.

Ansätze zur Behandlung der hier beschriebenen Optimierungsprobleme, an deren Ausarbeitung der Verfasser beteiligt war, finden sich in [172]. Die dort beschriebene Problemlösung wird hier nur teilweise aufgegriffen, weil sie einige Schwächen enthält. Zu ihnen gehören die reellwertige Behandlung des ganzzahligen Problems, die praxisferne Verwendung unscharfer Präsenzwahrscheinlichkeiten (das heißt unscharfer Varianten der $p_i(t)$ aus Satz 3.12) und eine problematische Schärfung der unscharfen Wahrscheinlichkeiten. Der Ansatz zur Lösung des dritten Optimierungsproblems im Abschnitt 4.2.4 wurde weitgehend aus [172] übernommen.

Bei den nachfolgend betrachteten Akquisen wird davon ausgegangen, dass sie von *endlicher* Dauer sind, $t_{\text{end}} < \infty$ (siehe Def. 3.11), das bedeutet, die zeitlichen Punktwahrscheinlichkeiten des Knotenwechsels $\varphi_{ij}(t)$ (siehe Def. 3.9) besitzen endliche Träger (siehe Satz 3.16).

4.1 Behandlung der Unschärfe

Zur Konzentration auf die Behandlung der unscharfen Kapazitätsschranken wird zunächst angenommen, dass das Kundenverhalten für den betrachteten Akquiseplan

$$AP = (\underline{A}, \underline{R})$$

deterministisch ist. Im Abschnitt 4.1.1 werden dafür die Grundlagen gelegt. Darauf baut im Abschnitt 4.1.2 die Formulierung des ersten Optimierungsproblems auf.

4.1.1 Grundlagen

Die Annahme eines deterministischen Kundenverhaltens im Akquiseplan $AP = (\underline{A}, \underline{R})$ heißt hier, dass für jede Kundenanzahl $c_0 \in \mathbb{N}_0$, die dem Akquiseplan zum Zeitpunkt $t = 0$ zuge-

führt wird, bereits im Zeitpunkt $t = 0$ für den gesamten Verlauf der Akquise angegeben werden kann, wieviele Kunden sich zu den betrachteten Zeitpunkten $t = 0, 1, \dots, t_{\text{end}}$ jeweils in den Knoten $A_i \in \underline{A}$, $i = 1, \dots, N$ des Akquiseplans befinden. Dabei sei t_{end} so groß, dass sich alle Kunden zu diesem Zeitpunkt entweder im Knoten A_N oder im Knoten A_{N-1} befinden. Im Satz 3.16 wird gezeigt, unter welchen Voraussetzungen t_{end} endlich ist. Die Kundenanzahl in den Knoten wird nachfolgend definiert.

Definition 4.1: *Kundenanzahl zur Zeit t im Knoten A_i für c_0 Anfangskunden, Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden.*

Sei $AP = (\underline{A}, \underline{R})$ ein Akquiseplan und bezeichne t_0 den Beginn der Akquise und t_{end} die Dauer der Akquise gemäß Def. 3.11. Dann bezeichnet

$$z(c_0, i, t) \in \mathbb{N}_0$$

die *Kundenanzahl zur Zeit t im Knoten A_i für c_0 Anfangskunden*. Das ist die Anzahl der Kunden, die sich zum Zeitpunkt t , $t = 0, 1, \dots, t_{\text{end}}$ im Knoten A_i , $i = 1, \dots, N$ befinden, wenn die Akquise zum Zeitpunkt $t_0 = 0$ mit c_0 Kunden, $c_0 \in \mathbb{N}_0$ begonnen wurde. Und

$$\Delta^+ z(c_0, i, t) \in \mathbb{N}_0$$

bezeichnet die *Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden*. Das ist die Anzahl der Kunden, die sich zum Zeitpunkt t , $t = 0, 1, \dots, t_{\text{end}}$ im Knoten A_i , $i = 1, \dots, N$ befinden und die sich zum Zeitpunkt $t - 1$ nicht in diesem Knoten befunden haben, wenn die Akquise zum Zeitpunkt $t_0 = 0$ mit c_0 Kunden, $c_0 \in \mathbb{N}_0$ begonnen wurde. \square

Anmerkungen

1. Die Definition ist unabhängig von der Art des Kundenverhaltens. Letzteres kann stochastisch oder deterministisch sein. In jedem Fall kann zu festgesetzten diskreten Zeitpunkten während der Akquise in jedem Knoten des Akquiseplans beobachtet werden, wie viele Kunden sich dort befinden.
2. Die Anzahl neuer Kunden zu einem bestimmten Zeitpunkt in einem bestimmten Knoten kann nicht größer als die Anzahl aller Kunden zu diesem Zeitpunkt im betrachteten Knoten sein, das heißt

$$\Delta^+ z(c_0, i, t) \leq z(c_0, i, t) \quad (4.1)$$

für alle $t = 0, 1, \dots, t_{\text{end}}$, $i = 1, \dots, N$ und beliebige $c_0 \in \mathbb{N}_0$.

3. Weil jeder Akquiseplan genau die eine Quelle A_1 besitzt, gilt für den Beginn der Akquise

$$z(c_0, 1, 0) = c_0 \text{ bzw. } \Delta^+ z(c_0, 1, 0) = c_0 \quad (4.2)$$

und damit

$$z(c_0, i, 0) = 0 \text{ bzw. } \Delta^+ z(c_0, i, 0) = 0 \text{ für } i \neq 1 \quad (4.3)$$

für beliebige $c_0 \in \mathbb{N}_0$.

4. Weil jeder Akquiseplan genau zwei Senken A_{N-1} und A_N besitzt, gilt für das Ende der Akquise

$$z(c_0, N-1, t_{\text{end}}) + z(c_0, N, t_{\text{end}}) = c_0 \text{ für beliebige } c_0 \in \mathbb{N}_0. \quad (4.4)$$

5. Allgemein gilt für jeden Zeitpunkt t , $t = 0, 1, \dots, t_{\text{end}}$

$$\sum_{i=1}^N z(c_0, i, t) = c_0 \quad (4.5)$$

und infolge (4.1):

$$\sum_{i=1}^N \Delta^+ z(c_0, i, t) \leq c_0 \quad (4.6)$$

für beliebige $c_0 \in \mathbb{N}_0$.

6. Kapazitätsschranken hängen nicht von der Zeit t ab.
7. In den Gleichungen (4.1) – (4.6) gibt t sowohl den Zeitpunkt der Betrachtung des Akquiseplans als auch die Dauer der Akquise für die betrachteten Kunden an. Diese Übereinstimmung tritt nur auf, wenn alle Kunden gleichzeitig dem Akquiseplan zugeführt werden. Sie gilt nicht mehr, wenn Kunden zu unterschiedlichen Zeitpunkten in den Akquiseplan gegeben werden, wie in der Fragestellung des zweiten Optimierungsproblems.
8. Wie bereits in den *Anmerkungen zu Annahmen und Bezeichnungsweisen* auf Seite 51 hingewiesen wurde, wird über einen Kunden, auf den eine Akquiseaktion angewendet worden ist gesagt, er befinde sich in dem der Aktion entsprechenden Knoten. Solange er auf die Akquiseaktion nicht reagiert, bleibt er dort.

Zunächst werden in Anlehnung an [99, 138] unscharfe Kapazitätsschranken definiert.

Definition 4.2: *Unschärfe Kapazitätsschranke.*

Eine *unscharfe Kapazitätsschranke* l ist eine unscharfe Menge über \mathbb{N}_0 , $l \in \mathcal{IF}(\mathbb{N}_0)$, deren Zugehörigkeitsfunktion, $\mu_l : \mathbb{N}_0 \rightarrow [0, 1]$ monoton abnehmend ist und für die gilt:

Entweder ist $\mu_l \equiv 1$, das heißt $\mu_l(x) = 1$ für alle $x \in \mathbb{N}_0$ oder es gibt $a, b \in \mathbb{N}_0$, so dass $\mu_l(x) = 1$ für $x \leq a$ und $\mu_l(x) = 0$ für $x > b$. \square

Anmerkungen

1. Der Sonderfall $a + 1 = b$ ist in der Definition eingeschlossen. In diesem Fall ist $(a, b) \cap \mathbb{N}_0 = \emptyset$.
2. Die Kapazitätsschranken l und ihre Parameter a, b werden entsprechend den Knoten des Akquiseplans indiziert, also $l_i \in \mathbb{IF}(\mathbb{N}_0)$ und $a_i, b_i \in \mathbb{N}_0$ für $i = 1, \dots, N$.
3. Unschärfe Kapazitätsschranken mit $\mu_l \equiv 1$ stellen unbeschränkte Kapazitäten dar.
4. Den Knoten A_1, A_{N-1} und A_N entsprechen keine wirklichen Akquiseaktionen. Sie markieren die Quelle bzw. die beiden Senken des Akquiseplans. Ihre Kapazitäten werden als unbeschränkt angenommen. Deshalb gilt $\mu_{l_i} \equiv 1$ für $l_i \in \mathbb{IF}(\mathbb{N}_0)$ mit $i \in \{1, N-1, N\}$.
5. Den Knoten $A_i, i \notin \{1, N-1, N\}$ entsprechen wirkliche Akquiseaktionen. Ihre Kapazitäten sind beschränkt. Deshalb gilt für $l_i \in \mathbb{IF}(\mathbb{N}_0), i = 2, 3, \dots, N-2$:

Es gibt $a_i, b_i \in \mathbb{N}_0, a_i, b_i < \infty$, so dass

$$\mu_{l_i}(x) = 1 \text{ für } x \leq a_i \text{ und } \mu_{l_i}(x) = 0 \text{ für } x \geq b_i \quad (4.7)$$

und, sofern $(a_i, b_i) \cap \mathbb{N}_0 \neq \emptyset$,

$$\mu_{l_i}(x) \text{ für } a_i < x < b_i \text{ monoton abnehmend ist.} \quad (4.8)$$

Für unscharfe Kapazitätsschranken mit nicht-konstanter Zugehörigkeitsfunktion wird folgende Umkehrabbildung eingeführt, die bei der Formulierung der Optimierungsprobleme nützlich ist.

Definition 4.3: Verallgemeinerte Umkehrabbildung einer monoton abnehmenden Zugehörigkeitsfunktion über \mathbb{N}_0 .

Sei $\mu_l : \mathbb{N}_0 \rightarrow [0, 1]$ eine monoton abnehmende Zugehörigkeitsfunktion. Dann wird

$$\mu_l^{-1} : (0, 1] \rightarrow \mathbb{N}_0 \text{ mit } \mu_l^{-1}(y) := \max\{x \in \mathbb{N}_0 \mid \mu_l(x) \geq y\} \quad (4.9)$$

die verallgemeinerte Umkehrabbildung von μ_l genannt. \square

Anmerkungen

1. Der Wertebereich der verallgemeinerten Umkehrabbildung μ_l^{-1} ist der Träger von l , $\text{suppl } l$.
2. Der Wert $\mu_l^{-1}(y)$ für $y \in (0, 1]$ ist das Maximum des y -Schnittes ${}^y l$ (vgl. Def. 2.5).
3. Die Umkehrabbildung besitzt im Allgemeinen weder die Eigenschaft $\mu_l^{-1}(\mu_l(x)) = x$ für alle $x \in \mathbb{N}_0$ noch $\mu_l(\mu_l^{-1}(y)) \equiv y$ für alle $y \in (0, 1]$. Ihre Eigenschaften sind auf die Opti-

mierungsprobleme abgestimmt und werden bei der Formulierung derselben verständlich. Eine Vorstellung vermitteln die Beispiele in Tabelle 4.1 für die in Abb. 4.1 gezeigte unscharfe Kapazitätsbeschränkung.

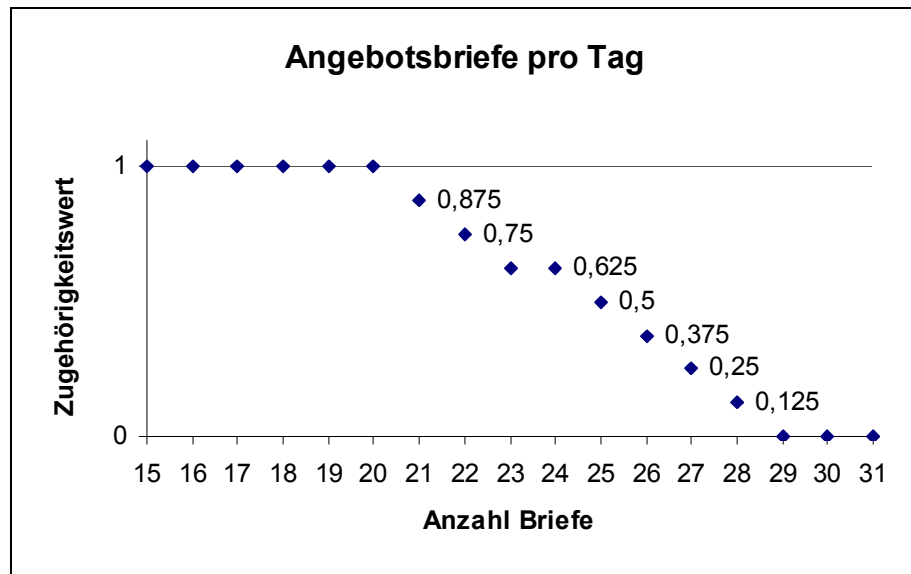


Abbildung 4.1: Beispiel einer unscharfen Kapazitätsschranke. Die Zahlenwerte geben den Zugehörigkeitswert an.

$y \in (0, 1]$	$x = \mu_1^{-1}(y) \in \mathbb{N}_0$
0,1	28
0,3	26
0,5	25
0,6	24
0,625	24
0,9	20
1	20

Tabelle 4.1: Werte der Umkehrabbildung für die Zugehörigkeitsfunktion in Abb. 4.1.

4. Wenn die verallgemeinerte Umkehrabbildung auf das Bild des Trägers von 1 unter μ_1 beschränkt wird, das ist die Menge $\mu_1(\text{suppl}) := \{y \in (0, 1] \mid \exists x \in \text{suppl} : \mu_1(x) = y\}$, und wenn die Zugehörigkeitsfunktion μ_1 streng monoton abnehmend ist, dann stimmt die verallgemeinerte Umkehrabbildung mit der echten Umkehrabbildung von μ_1 überein.

Für die Darstellung von unscharfen Nebenbedingungen gibt es in der Literatur viele Ansätze. Alle orientieren sich an der Form der Nebenbedingungen eines gewöhnlichen scharfen linearen Optimierungsproblems,

$$A \cdot x \leq b$$

für passend gewählte Matrix A , Vektor mit den Optimierungsvariablen x und rechte Seite b . Die Unschärfe der Nebenbedingungen erfordert die geeignete Definition der kleiner-gleich-Relation. Einige Ansätze ersetzen sie durch eine unscharfe oder scharfe Inklusion zwischen unscharfen Mengen, andere betrachten so genannte α -Schnitte (Def. 2.5) der Seiten rechts und links der kleiner-gleich-Relation, um sie auf die gewöhnliche kleiner-gleich-Relation zwischen scharfen Zahlen zurückzuführen [103]. Hier wird ein anderer Weg gewählt, der sich der unscharfen Element-Relation aus Kapitel 2 bedient: Für $A \in \mathbb{IF}(X)$ und $x \in X$ symbolisiert

$$x \varepsilon A$$

die unscharfe Element-Relation für die unscharfe Menge A und ein Element x ihrer Grundmenge. Für ihren Wahrheitswert $\llbracket x \varepsilon A \rrbracket$ gilt

$$\llbracket x \varepsilon A \rrbracket = \mu_A(x). \quad (4.10)$$

Dieser Zusammenhang wird nachfolgend für die Formulierung der Optimierungsprobleme benutzt.

4.1.2 Erstes Optimierungsproblem

Das erste Optimierungsproblem lautet (vgl. Abschnitt 3.4):

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan gleichzeitig unter Einhaltung der Kapazitätsschranken der Akquiseaktionen zugeführt werden kann?

Daraus folgen die Zielfunktion

$$c_0 \stackrel{!}{=} \max$$

und, unter der Annahme deterministischen Kundenverhaltens, die unscharfen Nebenbedingungen

$$\Delta^+ z(c_0, i, t) \varepsilon l_i \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}} \quad (4.11)$$

für $z(c_0, i, t)$, $i = 1, \dots, N$, $t = 0, 1, \dots, t_{\text{end}}$ gemäß Def. 4.1.

Wegen (4.10) folgt aus (4.11):

$$\llbracket \Delta^+ z(c_0, i, t) \varepsilon l_i \rrbracket = \mu_{l_i}(\Delta^+ z(c_0, i, t)) \quad (4.12)$$

für $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$. Aufgrund der Eigenschaften der unscharfen Kapazitätsschranken l_i (vgl. Def. 4.2) gilt für den Wahrheitswert (4.12):

$$\llbracket \Delta^+ z(c_0, i, t) \in I_i \rrbracket \begin{cases} = 1 & \text{für } \Delta^+ z(c_0, i, t) \leq a_i \\ \in (0, 1) & \text{für } a_i < \Delta^+ z(c_0, i, t) < b_i \\ = 0 & \text{für } b_i \leq \Delta^+ z(c_0, i, t) \end{cases} \quad (4.13)$$

Die Gleichung 4.13 deutet bereits an, dass das erste Optimierungsproblem unter Annahme deterministischen Kundenverhaltens für einen *vorgegebenen* Wahrheitswert zwischen null und eins ein *lineares Optimierungsproblem* ist. Welche Wahrheitswerte sollen aber in (4.12) betrachtet werden? In der Literatur, z. B. in [99] findet sich häufig der Ansatz, für das Zutreffen der Nebenbedingungen ein Mindestniveau zu verlangen. Dem entspricht hier ein Mindestwahrheitswert $\delta \in (0, 1]$, dem alle Nebenbedingungen genügen müssen und der im Folgenden *Wahrheitsniveau* genannt wird:

$$\llbracket \Delta^+ z(c_0, i, t) \in I_i \rrbracket = \mu_{I_i}(\Delta^+ z(c_0, i, t)) \geq \delta$$

für ein gegebenes $\delta \in (0, 1]$ und alle $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$. Dieser Ansatz wird in folgender Definition verallgemeinert.

Definition 4.4: *Scharfes Äquivalent eines linearen Optimierungsproblems mit unscharfer rechter Seite zum Wahrheitsniveau δ .*

Gegeben sei ein lineares Optimierungsproblem mit beliebiger Zielfunktion cx und den Nebenbedingungen

$$Ax \in b,$$

mit $c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ und $b \in [\text{IF}(\mathbb{R})]^m$ sowie ein vorgegebener Wahrheitswert $\delta \in (0, 1]$, der als *Wahrheitsniveau* bezeichnet wird. Dann bezeichnet das Optimierungsproblem mit derselben Zielfunktion cx und den Nebenbedingungen

$$\llbracket Ax \in b \rrbracket \geq \delta$$

das *scharfe Äquivalent* des ursprünglichen *Optimierungsproblems mit unscharfer rechter Seite zum Wahrheitsniveau δ* . \square

Jetzt kann das Optimierungsproblem für deterministisches Kundenverhalten vollständig beschrieben werden.

Definition 4.5: *Erstes Optimierungsproblem zum Wahrheitsniveau δ für deterministisches Kundenverhalten.*

Gegeben sei ein Akquiseplan AP , dem einmalig zum Zeitpunkt $t = 0$ die Anzahl $c_0 \in \mathbb{N}_0$ an Kunden zugeführt wird. Das Kundenverhalten sei deterministisch, so dass die Kundenanzahl $z(c_0, i, t) \in \mathbb{N}_0$ und die Anzahl neuer Kunden

$\Delta^+z(c_0, i, t) \in \mathbb{N}_0$ zum Zeitpunkt $t = 0, 1, \dots, t_{\text{end}}$ im Knoten A_i , $i = 1, \dots, N$ bereits zum Zeitpunkt $t = 0$ bekannt sei. Der Endzeitpunkt t_{end} sei so gewählt, dass $z(c_0, N-1, t_{\text{end}}) + z(c_0, N, t_{\text{end}}) = c_0$. Sei δ ein fest vorgegebener Wahrheitswert aus dem Intervall $(0, 1]$, der *Wahrheitsniveau* genannt wird.

Dann lautet das *erste Optimierungsproblem zum Wahrheitsniveau δ für deterministisches Kundenverhalten*:

$$c_0 \stackrel{!}{=} \max$$

unter den Nebenbedingungen

$$\llbracket \Delta^+z(c_0, i, t) \in l_i \rrbracket = \mu_{l_i}(\Delta^+z(c_0, i, t)) \geq \delta \quad (4.14)$$

für alle $i = 1, \dots, N$ und alle $t = 0, 1, \dots, t_{\text{end}}$. □

Anmerkung

Die Definition schließt $\delta = 0$ aus, weil für $\llbracket \Delta^+z(c_0, i, t) \in l_i \rrbracket = 0$ die Anzahl neuer Kunden zum Zeitpunkt t im Knoten A_i beliebig sein könnte. Für unbeschränkte Kapazitäten, wie sie in den Knoten A_1, A_{N-1}, A_N vorliegen, sind jedoch unscharfe Kapazitätsbeschränkungen mit $\mu \equiv 1$ vorgesehen. Außerdem enthielte die Aussage $\Delta^+z(c_0, i, t) \in l_i$ für $\delta = 0$ keine Information, weil sie wegen $\llbracket \Delta^+z(c_0, i, t) \in l_i \rrbracket \geq 0$ auch falsch sein dürfte.

Unter Ausnutzung der Eigenschaften unscharfer Kapazitätsschranken kann das erste Optimierungsproblem als ganzzahliges Optimierungsproblem gelöst werden. Dies ist Inhalt des nächsten Satzes.

Satz 4.1: Das erste Optimierungsproblem zum Wahrheitsniveau δ mit deterministischem Kundenverhalten aus Def. 4.5 ist ein *ganzzahliges Optimierungsproblem* und lautet:

$$c_0 \stackrel{!}{=} \max$$

unter den Nebenbedingungen $c_0 \in \mathbb{N}_0$ und

$$\Delta^+z(c_0, i, t) \leq \mu_{l_i}^{-1}(\delta) \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}. \quad (4.15)$$

□

Beweis: Weil die Zielfunktion im Satz mit der Zielfunktion in Def. 4.5 übereinstimmt, ist nur zu zeigen, dass die Menge der zulässigen Punkte im Satz mit jener in der Definition identisch ist.

Aus $\llbracket \Delta^+ z(c_0, i, t) \in l_i \rrbracket = \mu_{l_i}(\Delta^+ z(c_0, i, t)) \geq \delta_{l_i}$ folgt

$$\Delta^+ z(c_0, i, t) \in \{x \in \mathbb{N}_0 \mid \mu_{l_i}(x) \geq \delta_{l_i}\}$$

und damit

$$\Delta^+ z(c_0, i, t) \leq \max\{x \in \mathbb{N}_0 \mid \mu_{l_i}(x) \geq \delta_{l_i}\} = \mu_{l_i}^{-1}(\delta_{l_i}).$$

Umgekehrt folgt aus $\Delta^+ z(c_0, i, t) \leq \mu_{l_i}^{-1}(\delta) = \max\{x \in \mathbb{N}_0 \mid \mu_{l_i}(x) \geq \delta_{l_i}\}$ und aus der Eigenschaft, dass μ_{l_i} eine monoton abnehmende Funktion ist: $\mu_{l_i}(\Delta^+ z(c_0, i, t)) \geq \delta_{l_i}$, also

$$\llbracket \Delta^+ z(c_0, i, t) \in l_i \rrbracket \geq \delta.$$

Damit sind die Mengen der zulässigen Punkte im Satz und in der Definition identisch. Wegen $c_0 \in \mathbb{N}_0$ ist das Optimierungsproblem ganzzahlig. ■

Anmerkung zu Satz 4.1

Das Optimierungsproblem ist linear, weil $\Delta^+ z(c_0, i, t)$ zu jedem Zeitpunkt $t = 0, 1, \dots, t_{\text{end}}$ in jedem Knoten $i = 1, \dots, N$ ein bestimmtes Vielfaches der Optimierungsvariablen c_0 beträgt.

Das erste Optimierungsproblem wurde dargestellt, um beispielhaft die Behandlung von Unschärfe zu erklären. Die Bildung des *scharfen Äquivalents* kann völlig analog auf das zweite und das dritte Optimierungsproblem übertragen werden. Der nächste Abschnitt knüpft an die vorigen Ergebnisse an und erweitert sie auf stochastisches Kundenverhalten.

4.2 Erwartungswertansatz und Optimierung mit zufälligen Nebenbedingungen

Für die Behandlung des tatsächlichen, also stochastischen Kundenverhaltens werden im ersten Unterabschnitt einige Grundlagen eingeführt. Darauf aufbauend werden im Abschnitt 4.2.2 zwei Ansätze für das erste Optimierungsproblem formuliert. Der Ansatz auf Grundlage von *Erwartungswerten* und der Ansatz aus dem Bereich *Optimierung mit zufälligen Nebenbedingungen* („chance constrained programming“¹⁾) werden auf das zweite Optimierungsproblem übertragen und im Abschnitt 4.2.3 dargestellt. Schließlich wird gezeigt, wie das dritte Optimierungsproblem durch iteratives Lösen des zweiten Optimierungsproblems, unabhängig von dem gewählten Lösungsansatz, gelöst werden kann (siehe Abschnitt 4.2.4).

1. Auch in der deutschen Literatur ist die englische Bezeichnung „chance constrained programming“ geläufiger als die deutsche Übersetzung. Daneben wird die Bezeichnung „probabilistic constrained programming“ teilweise mit gleicher Bedeutung, teilweise für spezielle Probleme des „chance constrained programming“ verwendet [108].

Wie im Abschnitt 4.1 wird von einem Akquiseplan $AP = (\underline{A}, \underline{R})$ mit Knoten $A_i \in \underline{A}$ und Kanten aus \underline{R} ausgegangen, $i = 1, \dots, N$. Wie immer sei A_1 die Quelle und A_{N-1} und A_N seien die Senken des Akquiseplans. Für jeden Knoten gebe es eine unscharfe Kapazitätsschranke l_i , wobei l_1 , l_{N-1} und l_N als unbeschränkt angenommen werden.

4.2.1 Grundlagen

Im Gegensatz zum Abschnitt 4.1 sind in diesem Abschnitt $z(c_0, i, t)$ und $\Delta^+ z(c_0, i, t)$ (vgl. Def. 4.1) Realisierungen der stochastischen Größen $\zeta(c_0, i, t)$ bzw. $\Delta^+ \zeta(c_0, i, t)$, jeweils für $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$, also in der Regel nicht zum Zeitpunkt $t = 0$ bekannt.

Im Kapitel 3 zeigt Satz 3.12, dass die Wahrscheinlichkeiten $p_i(t) := W\{A(i, t)\}$, $i = 1, \dots, N$, $t = 0, 1, \dots, t_{\text{end}}$ für den dort angegebenen Wahrscheinlichkeitsraum (vgl. Def. 3.4) iterativ berechnet werden können,

$$p(t+1) = p(t)P_{\omega}(t+1) = p(0) \prod_{\tau=1}^{t+1} P_{\omega}(\tau) .$$

Offensichtlich gilt $W\{A(i, t)\} = p_i(t) = W\{\zeta(1, i, t) = 1\}$. Weiter unten wird gezeigt, wie daraus weitere Wahrscheinlichkeiten für das Auftreten bestimmter Anzahlen neuer Kunden, $\zeta(c_0, i, t)$ berechnet werden können. Für Aussagen über die Zufallsgröße $\Delta^+ \zeta(c_0, i, t)$ ist es nötig, Satz 3.12 zu erweitern und zuvor, anknüpfend an die Def. 3.6 im Abschnitt 3.2.2.1, eine neue Präsenzwahrscheinlichkeit einzuführen.

Definition 4.6: *Präsenzwahrscheinlichkeit neuer Kunden, Vektor der Präsenzwahrscheinlichkeiten neuer Kunden.*

Die Wahrscheinlichkeit $\Delta^+ p_j(t) = W\{A(j, t) \cap \bar{A}(j, t-1)\}$, $t = 0, 1, \dots, t_{\text{end}}$ mit der Festlegung

$$A(j, 0) \cap \bar{A}(j, -1) := A(j, 0) \tag{4.16}$$

(vgl. (3.3) in Def. 3.5) ist die *Präsenzwahrscheinlichkeit neuer Kunden*, das heißt die Wahrscheinlichkeit, dass ein Kunde zum Zeitpunkt t neu im Knoten A_j , $j = 1, \dots, N$ ist, also die Wahrscheinlichkeit, dass er sich zum Zeitpunkt $t-1$ in einem Knoten A_i , $i \neq j$ befindet und zum Zeitpunkt t im Knoten A_j . Diese Präsenzwahrscheinlichkeiten werden im *Vektor der Präsenzwahrscheinlichkeiten neuer Kunden* zum Zeitpunkt t ,

$$\Delta^+ p(t) = (\Delta^+ p_1(t), \dots, \Delta^+ p_N(t))$$

zusammengefasst. \square

Satz 4.2: Die Vektoren der Präsenzwahrscheinlichkeiten neuer Kunden $\Delta^+ p(t)$ gemäß Def. 4.6 können für $t = 0, 1, \dots, t_{\text{end}} - 1$ iterativ durch Matrizenmultiplikation berechnet werden:

$$\Delta^+ p(t+1) = p(t) \cdot [P_\omega(t+1) - \text{diag} P_\omega(t+1)], \quad (4.17)$$

bzw.

$$\Delta^+ p_j(t+1) = \sum_{i=1}^{j-1} p_i(t) \Psi_{ij}^{t+1}$$

für $j = 1, \dots, N$, wobei $p(t)$ der Vektor der Präsenzwahrscheinlichkeiten zum Zeitpunkt t nach Def. 3.6 ist und $P_\omega(t+1) = (\Psi_{ij}^{t+1})_{i,j=1}^N$ eine Matrix gemäß Satz 3.12. \square

Beweis: Aus Def. 3.6 ist bekannt, dass

$$p_i(t) := W\{A(i, t)\} \quad (4.18)$$

für $i = 1, \dots, N$ und $t = 0, 1, \dots$ und aus Satz 3.12, dass

$$\psi_{ij}^{t+1} = \sum_{\tau=0}^t \omega_i^t(\tau) \pi_{ij}^t(\tau) \quad (4.19)$$

für $i, j = 1, \dots, N$ und $t = 0, 1, \dots$ sowie aus den Definitionen 3.7 und 3.10, dass

$$\omega_i^t(\tau) = W\{A(i, t-\tau) \cap \bar{A}(i, t-\tau-1) \mid A(i, t)\} \quad (4.20)$$

und

$$\pi_{ij}^t(\tau) = W\{A(j, t+1) \mid A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}, \quad (4.21)$$

jeweils für $i, j = 1, \dots, N$, $t = 0, 1, \dots$, $\tau = 0, 1, \dots, t$ und unter Beachtung von (4.16).

Nach Einsetzen von (4.20) und (4.21) in (4.19) ist mit (4.18):

$$\begin{aligned} \sum_{i=1}^{j-1} p_i(t) \Psi_{ij}^{t+1} &= \sum_{i=1}^{j-1} \left[W\{A(i, t)\} \sum_{\tau=0}^t (W\{A(i, t-\tau) \cap \bar{A}(i, t-\tau-1) \mid A(i, t)\} \right. \\ &\quad \cdot W\{A(j, t+1) \mid A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\}) \Big]. \end{aligned}$$

Nach Anwendung der Rechenregeln für bedingte Wahrscheinlichkeiten und unter Berücksichtigung von Eigenschaft 3 des Wahrscheinlichkeitsraums (M, E, W) (vgl. Def. 3.4) folgt:

$$\begin{aligned} \sum_{i=1}^{j-1} p_i(t) \Psi_{ij}^{t+1} &= \sum_{i=1}^{j-1} \sum_{\tau=0}^t W\{A(j, t+1) \cap A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)\} \\ &= \sum_{i=1}^{j-1} W\left\{A(j, t+1) \cap \bigcup_{\tau=0}^t [A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)]\right\}. \end{aligned}$$

Im Beweis von Hilfssatz 3.3 wurde gezeigt (vgl. (3.8)), dass

$$\bigcup_{\tau=0}^t [A(i, t) \cap A(i, t-\tau) \cap \bar{A}(i, t-\tau-1)] = A(i, t),$$

und damit und mit $W\left\{A(j, t+1) \cap \bigcup_{i=j+1}^N A(i, t)\right\} = 0$ ist

$$\begin{aligned} \sum_{i=1}^{j-1} p_i(t) \Psi_{ij}^{t+1} &= \sum_{i=1}^{j-1} W\{A(j, t+1) \cap A(i, t)\} \\ &= W\left\{A(j, t+1) \cap \bigcup_{i=1}^{j-1} A(i, t)\right\} + W\left\{A(j, t+1) \cap \bigcup_{i=j+1}^N A(i, t)\right\} \\ &= W\left\{A(j, t+1) \cap \bigcup_{i \neq j} A(i, t)\right\} = W\{A(j, t+1) \cap \bar{A}(j, t)\} \end{aligned}$$

Mit Def. 4.6 ist alles gezeigt. ■

Nachdem die Wahrscheinlichkeit $\Delta^+ p_i(t)$ definiert und eine Vorschrift zu ihrer Berechnung eingeführt worden ist, werden jetzt die eingangs erwähnten Zufallsgrößen eingeführt.

Definition 4.7: *Zufallsgröße der Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden.*

Sei $AP = (\underline{A}, \underline{R})$ ein Akquiseplan und bezeichne $t = 0$ den Beginn der Akquise und t_{end} das Akquiseende. Dann bezeichnet

$$\Delta^+ \zeta(c_0, i, t)$$

die Zufallsgröße der Anzahl neuer Kunden zur Zeit t im Knoten A_i für c_0 Anfangskunden, $t = 0, 1, \dots, t_{\text{end}}$, $i = 1, \dots, N$, $c_0 \in \mathbb{N}_0$, wenn sie die Eigenschaft

$$W\{\Delta^+\zeta(1, i, t) = 1\} = \Delta^+p_i(t)$$

für alle $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$ besitzt, wobei $\Delta^+p_i(t)$ die Präsenzwahrscheinlichkeit neuer Kunden nach Def. 4.6 ist, die gemäß Satz 4.2 berechnet werden kann. \square

Anmerkungen

1. Weil jeder Akquiseplan genau die eine Quelle A_1 besitzt, gilt für den Beginn der Akquise

$$W\{\Delta^+\zeta(1, 1, 0) = 1\} = \Delta^+p_1(0) = 1 \quad (4.22)$$

und damit

$$W\{\Delta^+\zeta(1, i, 0) = 1\} = \Delta^+p_i(0) = 0 \text{ für } i \neq 1. \quad (4.23)$$

2. Aus (4.17) ist ersichtlich, dass

$$\sum_{i=1}^N W\{\Delta^+\zeta(1, i, t) = 1\} = \sum_{i=1}^N \Delta^+p_i(t) \leq 1 \text{ für } t = 0, 1, \dots, t_{\text{end}}. \quad (4.24)$$

3. In den Gleichungen (4.22) – (4.24) gibt t sowohl den Zeitpunkt der Betrachtung des Akquiseplans an als auch die Dauer der Akquise für die betrachteten Kunden an. Diese Übereinstimmung tritt nur auf, wenn alle Kunden gleichzeitig dem Akquiseplan zugeführt werden. Sie gilt nicht mehr, wenn Kunden zu unterschiedlichen Zeitpunkten in den Akquiseplan gegeben werden, wie in der Fragestellung des zweiten Optimierungsproblems.

Die Zufallsgröße $\Delta^+\zeta(1, i, t)$ kann nur die Werte 0 oder 1 annehmen. Dabei folgt sie für *gegebenes* $t \in \{0, 1, \dots, t_{\text{end}}\}$ und *festes* $i \in \{1, \dots, N\}$ einer diskreten Verteilung mit den Wahrscheinlichkeiten

$$W\{\Delta^+\zeta(1, i, t) = 1\} = \Delta^+p_i(t) \text{ und } W\{\Delta^+\zeta(1, i, t) = 0\} = 1 - \Delta^+p_i(t).$$

Dies ist eine diskrete Wahrscheinlichkeitsverteilung über die Werte 0 und 1, das heißt die Zufallsgröße $\Delta^+\zeta(1, i, t)$ folgt einer Alternativverteilung $A_{\Delta^+p_i(t)}$:

$$\Delta^+\zeta(1, i, t) \sim A_{\Delta^+p_i(t)}$$

[155]. Für die Formulierung der Optimierungsprobleme zur maximalen Beschickung von Akquiseplänen reicht das nicht aus. Dafür ist die Zufallsgröße $\zeta(c_0, i, t)$ für beliebiges $c_0 \in \mathbb{N}$ zu betrachten. Über die Wahrscheinlichkeitsverteilung

$$W\left\{\Delta^+\zeta(c_0, i, t) = z\right\} \text{ für } z = 0, 1, \dots, c_0$$

gibt der folgende Satz Auskunft.

Satz 4.3: Gegeben sei die Zufallsgröße $\Delta^+\zeta(c_0, i, t)$ gemäß Def. 4.7. Sie kann Werte $z \in \mathbb{N}$ zwischen 0 und c_0 annehmen. Dabei folgt sie für gegebenes $t \in \{0, 1, \dots, t_{\text{end}}\}$ und festes $i \in \{1, \dots, N\}$ einer diskreten Verteilung mit den Wahrscheinlichkeiten für die Realisierung z :

$$W\left\{\Delta^+\zeta(c_0, i, t) = z\right\} = \binom{c_0}{z} [\Delta^+p_i(t)]^z [1 - \Delta^+p_i(t)]^{c_0 - z},$$

also einer *Binomialverteilung* $B_{c_0, \Delta^+p_i(t)}$, das heißt

$$\zeta(c_0, i, t) \sim B_{c_0, \Delta^+p_i(t)}. \quad (4.25)$$

□

Beweis: Nach (4.25) folgt die Zufallsgröße für einen einzelnen Kunden einer Alternativverteilung. Der Weg eines Kunden durch den Akquiseplan wird durch eine Knotenfolge beschrieben (vgl. Def. 3.3). Knotenfolgen sind Elementarereignisse im Wahrscheinlichkeitsraum (M, E, W) und unabhängig voneinander (vgl. Anmerkung 1 zu Def. 3.6). Deshalb kann die Zufallsgröße $\Delta^+\zeta(c_0, i, t)$ als c_0 -faches Auftreten der Zufallsgröße $\Delta^+\zeta(1, i, t)$ betrachtet werden und ihr Wert als die *Anzahl* der Ereignisse $\Delta^+\zeta(1, i, t) = 1$. Dieses Ereignis folgt aber bekanntlich einer Binomialverteilung $B_{n, p}$, wobei der Parameter n gleich der Anzahl der hintereinander ausgeführten Alternativversuche ist, also c_0 , und der Parameter p ist gleich der Wahrscheinlichkeit des Ereignisses $\zeta(1, i, t) = 1$, also $\Delta^+p_i(t)$ [155]. Damit ist $\zeta(c_0, i, t) \sim B_{c_0, \Delta^+p_i(t)}$. ■

Anmerkungen zu Satz 4.3

1. Weil der Erwartungswert für die Binomialverteilung $B_{n, p}$ gleich $n \cdot p$ ist, gilt

$$E\Delta^+\zeta(c_0, i, t) = c_0 \cdot \Delta^+p_i(t) \quad (4.26)$$

für gegebenes $t \in \{0, 1, \dots, t_{\text{end}}\}$ und festes $i \in \{1, \dots, N\}$.

2. Die Annahme, dass sich zwei Kunden unabhängig verhalten, wird in der Praxis nicht immer zutreffen, z. B. dann nicht, wenn sich zwei beworbene Kunden kennen und gegenseitig aus-

tauschen. Bei Akquisen mit großer Kundenmenge werden solche Fälle nur einen geringen Anteil darstellen. Um das mathematische Modell nicht unnötig zu verkomplizieren, werden sie hier vernachlässigt.

Mit den Ergebnissen des Satzes wird im nächsten Abschnitt das erste Optimierungsproblem formuliert.

4.2.2 Erstes Optimierungsproblem

Den allgemeinen Ansatz für das erste Optimierungsproblem folgt aus der Formulierung in Satz 4.1 durch Ersetzen von $\Delta^+z(c_0, i, t)$ durch $\Delta^+\zeta(c_0, i, t)$. Es lautet dann:

$$c_0 \stackrel{!}{=} \max$$

unter den Nebenbedingungen $c_0 \in \mathbb{N}_0$, $c_0 < \infty$ und

$$\Delta^+\zeta(c_0, i, t) \leq \mu_{l_i}^{-1}(\delta) \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}. \quad (4.27)$$

In dieser Formulierung ist die Behandlung der unscharfen Kapazitätsschranken durch Zurückführen auf den scharfen Fall gelöst. Wie ist aber in (4.27) die Ungleichung zwischen den Zufallsgrößen $\Delta^+\zeta(c_0, i, t)$ und den bestimmten Größen $\mu_{l_i}^{-1}(\delta)$ für $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$ zu verstehen? In der Literatur finden sich zahlreiche Ansätze dafür (vgl. z. B. [91]). Zwei davon werden in den beiden folgenden Unterabschnitten benutzt.

4.2.2.1 Erstes Optimierungsproblem als Erwartungswertproblem

Die einfachste Darstellung der Ungleichungen (4.27) folgt, wenn die Zufallsgröße $\Delta^+\zeta(c_0, i, t)$ durch ihren Erwartungswert $\mathbb{E}\Delta^+\zeta(c_0, i, t)$ ersetzt wird. Sein Wert ist gemäß (4.26) gleich $c_0 \cdot \Delta^+p_i(t)$. Damit lautet das Optimierungsproblem wie folgt:

Definition 4.8: *Erstes Optimierungsproblem als Erwartungswertproblem.*

Das erste Optimierungsproblem zum Wahrheitsniveau δ lautet als Erwartungswertproblem:

$$c_0 \stackrel{!}{=} \max \quad (4.28)$$

unter den Nebenbedingungen $c_0 \in \mathbb{N}_0$ und

$$c_0 \cdot \Delta^+p_i(t) \leq \mu_{l_i}^{-1}(\delta) \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}. \quad (4.29)$$

□

In dieser Gestalt besitzt das erste Optimierungsproblem eine explizite Lösung.

Satz 4.4: Das erste Optimierungsproblem als Erwartungswertproblem besitzt die Lösung

$$c_0 = \left\lfloor \min_{i=1, \dots, N} \left[\frac{\mu_{l_i}^{-1}(\delta)}{\max_{t=0, 1, \dots, t_{\text{end}}} \Delta^+ p_i(t)} \right] \right\rfloor, \quad (4.30)$$

wobei $\lfloor x \rfloor$, $x \in \mathbb{R}$ die größte ganze Zahl kleiner oder gleich x symbolisiert. \square

Beweis: Ohne Beschränkung der Allgemeinheit wird angenommen, dass jeder Knoten während der Akquise erreicht werden kann, das heißt

$$\exists t \in \{0, 1, \dots, t_{\text{end}}\} \exists i \in \{1, \dots, N\} : \Delta^+ p_i(t) \neq 0. \quad (4.31)$$

Andernfalls wäre im Akquiseplan $AP = (\underline{A}, \underline{R})$ eine Kante in \underline{R} mit Übergangswahrscheinlichkeit null vorhanden. Solche Kanten wären sinnlos und könnten entfernt werden, ohne die Kundenströme im Akquiseplan zu verändern. Der daraus resultierende Akquiseplan würde der Bedingung (4.31) genügen.

Weil für ein festes $i \in \{1, \dots, N\}$ die rechte Seite von (4.29) für jedes $t = 0, 1, \dots, t_{\text{end}}$ gleich ist, können die Nebenbedingungen knotenweise zusammengefasst werden:

$$c_0 \cdot \max_{t \in \{0, 1, \dots, t_{\text{end}}\}} \Delta^+ p_i(t) \leq \mu_{l_i}^{-1}(\delta) \text{ für } i = 1, \dots, N. \quad (4.32)$$

Wegen (4.31) kann für jedes $i = 1, \dots, N$ durch das Maximum auf der linken Seite der Nebenbedingungen geteilt werden:

$$c_0 \leq \frac{\mu_{l_i}^{-1}(\delta)}{\max_{t \in \{0, 1, \dots, t_{\text{end}}\}} \Delta^+ p_i(t)} \text{ für } i = 1, \dots, N.$$

Daraus folgt sofort:

$$c_0 \leq \min_{i=1, \dots, N} \left[\frac{\mu_{l_i}^{-1}(\delta)}{\max_{t \in \{0, 1, \dots, t_{\text{end}}\}} \Delta^+ p_i(t)} \right].$$

Weil $c_0 \in \mathbb{N}$ folgt aus der Zielfunktion (4.28) die Behauptung. \blacksquare

Anmerkungen zu Satz 4.4

1. Der Erwartungswert der Zufallsgröße $\Delta^+ \zeta(c_0, i, t)$ sagt nichts über ihre Varianz aus. Trotz Einhaltung der Nebenbedingungen (4.32) können die Realisierungen der Zufallsgröße, $\Delta^+ z(c_0, i, t)$ die Nebenbedingungen *beliebig stark* verletzen, das heißt

$$\Delta^+ z(c_0, i, t) > \mu_{l_i}^{-1}(\delta).$$

In der Praxis werden starke Überlastungen der Kapazitäten bei Akquiseaktionen nicht mit starken Unterlastungen der Kapazitäten wettgemacht. Genau genommen, kann nicht von *starken* Überlastungen gesprochen werden, sondern von *unzulässigen* Überlastungen. Denn starke Überlastungen sind bereits durch die unscharfen Kapazitätsschranken l_i möglich.

2. Die Problematik des Ansatzes wird offensichtlich beim Versuch, die Varianz der Zufallsgröße $\Delta^+ \zeta(c_0, i, t)$ einzubeziehen. Wegen ihrer Binomialverteilung (4.25) beträgt sie

$$\text{Var } \Delta^+ \zeta(c_0, i, t) = c_0 \cdot \Delta^+ p_i(t) \cdot (1 - \Delta^+ p_i(t))$$

für $i \in \{1, \dots, N\}$, $t = 0, 1, \dots, t_{\text{end}}$. Die in der Zielfunktion (4.28) geforderte Maximierung von c_0 führt unmittelbar zur Vergrößerung der Varianz und damit in der Regel zu unzulässigen Lösungen durch Verletzung der Nebenbedingungen.

3. Bei der Simulation in Abschnitt 3.3 tritt das zuvor beschriebene Problem nicht zutage, weil sie keine Monte-Carlo-Simulation ist, sondern nur Erwartungswerte berechnet.

Die Unzulänglichkeiten des ersten Optimierungsproblems als Erwartungswertproblem werden vom nächsten Ansatz berücksichtigt.

4.2.2.2 Optimierung mit zufälligen Nebenbedingungen

Während beim ersten Optimierungsproblem als Erwartungswertproblem keine Aussage möglich ist, wie häufig die Nebenbedingungen eingehalten oder verletzt werden, misst der Ansatz der *Optimierung mit zufälligen Nebenbedingungen* („chance constrained programming“⁽²⁾) das Risiko unzulässiger Nebenbedingungen *qualitativ*. Dazu wird eine Mindestwahrscheinlichkeit δ_ζ vorgegeben, mit der die Nebenbedingungen eingehalten, also die Lösung des Optimierungsproblems zulässig sein soll. Im Folgenden wird δ_ζ als Wahrscheinlichkeitsniveau bezeichnet. Zur Unterscheidung erhält der Mindestwahrscheinlichkeitswert für die unscharfen Kapazitätsschranken den Index 1, δ_1 . Für (4.27) sieht die Formulierung dann wie folgt aus:

$$W \left\{ \Delta^+ \zeta(c_0, i, t) \leq \mu_{l_i}^{-1}(\delta_1) \right\} \geq \delta_\zeta \text{ für } i = 1, \dots, N \text{ und } t = 0, 1, \dots, t_{\text{end}}. \quad (4.33)$$

Für die weitere Beschreibung der Nebenbedingungen wird die *Verteilungsfunktion* der Zufallsgröße $\Delta^+ \zeta(c_0, i, t)$ für gegebenes $i = 1, \dots, N$ und festes $t = 0, 1, \dots, t_{\text{end}}$ benötigt. Sie sei mit

2. Eine andere deutsche Bezeichnung für „chance constrained programming“ ist „Optimierung mit Wahrscheinlichkeitsrestriktionen“.

$$F_{\Delta^+\zeta(c_0, i, t)}$$

bezeichnet. Bekanntlich gilt

$$F_{\Delta^+\zeta(c_0, i, t)} : \mathbb{N}_0 \rightarrow [0, 1]$$

und wegen (4.25) ist

$$F_{\Delta^+\zeta(c_0, i, t)}(z) = \sum_{x=0}^z \binom{c_0}{x} [\Delta^+p_i(t)]^x [1 - \Delta^+p_i(t)]^{c_0-x}.$$

Mit Hilfe der Verteilungsfunktion können die Nebenbedingungen (4.33) kürzer formuliert werden:

$$F_{\Delta^+\zeta(c_0, i, t)}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für } i = 1, \dots, N \text{ und } t = 0, 1, \dots, t_{\text{end}}. \quad (4.34)$$

Die Nebenbedingungen (4.34) verlangen, dass $\mu_{l_i}^{-1}(\delta_l)$ ein δ_ζ -Quantil der Verteilungsfunktion $F_{\Delta^+\zeta(c_0, i, t)}$ ist, für $i = 1, \dots, N$ und $t = 0, 1, \dots, t_{\text{end}}$. Damit kann das erste Optimierungsproblem als *Optimierung mit zufälligen Nebenbedingungen* formuliert werden:

Definition 4.9: *Erstes Optimierungsproblem als Optimierungsproblem mit zufälligen Nebenbedingungen.*

Das *erste Optimierungsproblem* zum Wahrheitsniveau δ_l und zum Wahrscheinlichkeitsniveau δ_ζ lautet als *Optimierungsproblem mit zufälligen Nebenbedingungen*:

$$c_0 \stackrel{!}{=} \max$$

unter den Nebenbedingungen $c_0 \in \mathbb{N}_0$ und

$$F_{B_{\Delta^+\zeta(c_0, i, t)}}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für } i = 1, \dots, N \text{ und } t = 0, 1, \dots, t_{\text{end}},$$

wobei $F_{B_{\Delta^+\zeta(c_0, i, t)}}$ die Verteilungsfunktion der Binomialverteilung für gegebenes

$i \in \{1, \dots, N\}$ und festes $t = 0, 1, \dots, t_{\text{end}}$ ist. \square

Wie für das *erste Optimierungsproblem als Erwartungswertproblem* im Abschnitt 4.2.2.1 lässt sich auch für das *erste Optimierungsproblem als Optimierungsproblem mit zufälligen Nebenbedingungen* ein Lösungsweg angeben. Er wird im Satz 4.7 beschrieben. Zuvor sind noch die *verallgemeinerte inverse Verteilungsfunktion* zu definieren und in den Hilfssätzen 4.5 und 4.6 zwei wichtige Eigenschaften der Binomialverteilung zu zeigen.

Definition 4.10: *Verallgemeinerte inverse Verteilungsfunktion* [69].

Sei F_ζ die Verteilungsfunktion der Zufallsgröße ζ . Dann ist

$$F_\zeta^{-1}(y) := \inf\{x : F_\zeta(x) \geq y\}, y \in [0, 1]$$

die *verallgemeinerte inverse Verteilungsfunktion* von F_ζ . □

Anmerkung

Für das δ -Quantil der Verteilungsfunktion F_ζ , ζ_δ , gilt

$$F_\zeta(\zeta_\delta) \geq \delta \text{ und } F_\zeta^{-1}(\delta) = \zeta_\delta. \quad (4.35)$$

Hilfssatz 4.5: Gegeben sei eine Zufallsgröße, die einer Binomialverteilung folgt, $\zeta \sim B_{n,p}$, $n \in \mathbb{N} \setminus \{0\}$, $p \in (0, 1)$. Die Verteilungsfunktion F_ζ ist in p streng monoton abnehmend. □

Beweis: [74] zeigt, dass für

$$f(p) := 1 - F_\zeta(k-1) = \sum_{\nu=k}^n \binom{n}{\nu} p^\nu (1-p)^{n-\nu} \quad (4.36)$$

mit $0 < k \leq n$ die Ableitung nach p für $p \in (0, 1)$ existiert und dass

$$\frac{d}{dp} f(p) = n \binom{n-1}{k-1} p^{k-1} (1-p)^{n-k}. \quad (4.37)$$

gilt. Da $n > 0$, der Binomialkoeffizient $\binom{n-1}{k-1} \geq 1$, p und $1-p$ und damit auch ihre Potenzen alle positiv sind, ist die Ableitung (4.37) ebenfalls positiv. Damit ist $f(p)$ in p streng monoton zunehmend und somit $F_\zeta(k-1)$ aus (4.36) in p streng monoton abnehmend. ■

Beispiel 4.1: Ein Beispiel für den Hilfssatz 4.5 zeigt Abb. 4.2. Dort sind die Binomialverteilungen für $n = 10$ und $p = 0,3$, $p = 0,5$ sowie $p = 0,7$ abgebildet. Offensichtlich ist der Wert der Verteilungsfunktion umso geringer, je größer die Wahrscheinlichkeit p ist.

Zur Vereinfachung der Formulierung wird nachfolgend die Verteilungsfunktion F_ζ der Zufallsvariablen $\zeta \sim B_{n,p}$ auch mit $F_{B_{n,p}}$ bezeichnet.

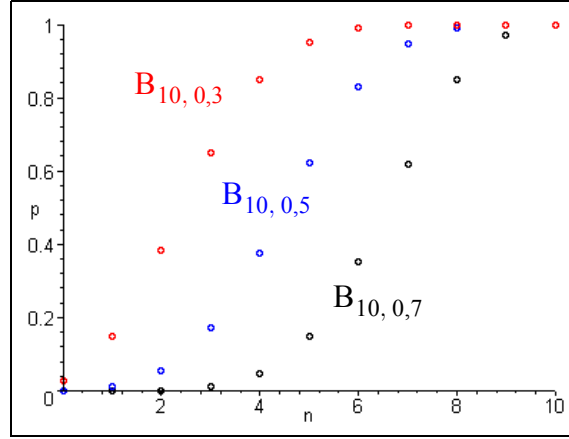


Abbildung 4.2: Beispiel für Hilfssatz 4.5: Die Verteilungsfunktionen der Binomialverteilungen $B_{10, 0.3}$, $B_{10, 0.5}$ und $B_{10, 0.7}$.

Hilfssatz 4.6: Gegeben sei eine Zufallsvariable ζ , die einer Binomialverteilung folgt, das heißt $\zeta \sim B_{n,p}$. Die Verteilungsfunktion $F_{B_{n,p}}$ ist in n schwach monoton abnehmend.

□

Beweis: Es ist zu zeigen, dass $F_{B_{n+1,p}} \leq F_{B_{n,p}}$ für $n \in \mathbb{N}_0$. Der Beweis nutzt die Identität

$$\binom{n+1}{\nu} = \binom{n}{\nu} + \binom{n}{\nu-1} \quad (4.38)$$

für $n \in \mathbb{N}_0$ und $\nu = 1, \dots, n$. Sei $\zeta \sim B_{n+1,p}$. Dann ist für $k < n+1$ mit (4.38)

$$\begin{aligned} F_{B_{n+1,p}}(k) &= W\{\zeta \leq k\} \\ &= \sum_{n=0}^k \binom{n+1}{\nu} p^\nu (1-p)^{n+1-\nu} = (1-p)^{n+1} + \sum_{n=1}^k \binom{n+1}{\nu} p^\nu (1-p)^{n+1-\nu} \\ &= (1-p)^{n+1} + \sum_{n=1}^k \left[\binom{n}{\nu} + \binom{n}{\nu-1} \right] p^\nu (1-p)^{n+1-\nu} \\ &= (1-p)^{n+1} + \sum_{n=1}^k \binom{n}{\nu} p^\nu (1-p)^{n+1-\nu} + \sum_{n=1}^k \binom{n}{\nu-1} p^\nu (1-p)^{n+1-\nu} \\ &= \sum_{n=0}^k \binom{n}{\nu} p^\nu (1-p)^{n+1-\nu} + \sum_{n=1}^k \binom{n}{\nu-1} p^\nu (1-p)^{n+1-\nu} \\ &= (1-p) \sum_{n=0}^k \binom{n}{\nu} p^\nu (1-p)^{n-\nu} + p \sum_{n=0}^{k-1} \binom{n}{\nu} p^\nu (1-p)^{n-\nu} \end{aligned}$$

$$= (1-p) \sum_{n=0}^k \binom{n}{\nu} p^{\nu} (1-p)^{n-\nu} + p \sum_{n=0}^k \binom{n}{\nu} p^{\nu} (1-p)^{n-\nu} - p \binom{n}{k} p^k (1-p)^{n-k}$$

$$= \sum_{n=0}^k \binom{n}{\nu} p^{\nu} (1-p)^{n-\nu} - p \binom{n}{k} p^k (1-p)^{n-k}$$

Da $p \binom{n}{k} p^k (1-p)^{n-k} \geq 0$, folgt

$$F_{B_{n+1,p}}(k) = \sum_{n=0}^k \binom{n+1}{\nu} p^{\nu} (1-p)^{n-\nu+1} \leq \sum_{n=0}^k \binom{n}{\nu} p^{\nu} (1-p)^{n-\nu} = F_{B_{n,p}}(k). \quad \blacksquare$$

Beispiel 4.2: Ein Beispiel für den Hilfssatz 4.6 zeigt Abb. 4.3. Dort sind die Binomialverteilungen für $p = 0,7$ und $n = 5$, $n = 10$ sowie $n = 15$ abgebildet. Offensichtlich ist der Wert der Verteilungsfunktion umso geringer, je größer die Anzahl n ist.

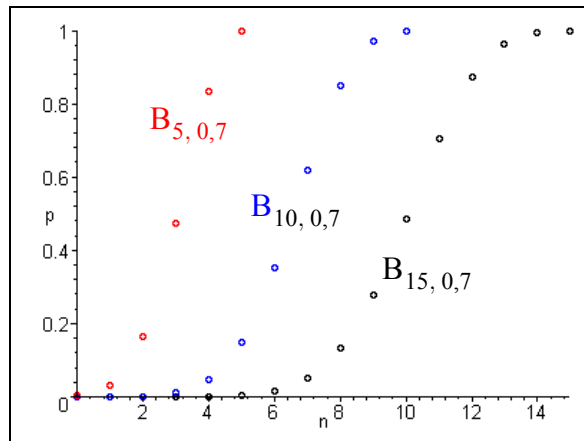


Abbildung 4.3: Beispiel für Hilfssatz 4.6: die Verteilungsfunktionen der Binomialverteilungen $B_{5,0.7}$, $B_{10,0.7}$ und $B_{15,0.7}$.

Auf Grundlage der beiden Hilfssätze wird der folgende Satz bewiesen, der den Lösungsweg des ersten Optimierungsproblems als *Optimierungsproblem mit zufälligen Nebenbedingungen* aufzeigt.

Satz 4.7: Das erste Optimierungsproblem als *Optimierungsproblem mit zufälligen Nebenbedingungen* besitzt die Lösung $c_0 \in \mathbb{N}_0$, die sich wie folgt berechnet:

1. Berechne $t_{\max}^i : \Delta^+ p_i(t_{\max}^i) = \max_{t \in \{0, 1, \dots, t_{\text{end}}\}} \Delta^+ p_i(t)$ für alle $i = 1, \dots, N$.

2. Berechne $c_0^i = \max \left\{ c_0' \mid F_{B_{c_0', \Delta^+ p_i(t_{\max}^i)}}^{-1}(\delta_c) = \mu_{l_i}^{-1}(\delta_l) \right\}$ für alle $i = 1, \dots, N$.

$$3. \text{ Berechne } c_0 = \min_{i \in \{1, \dots, N\}} c_0^i.$$

□

Beweis: Es ist zu zeigen, dass c_0 zulässig und optimal ist.

Zulässigkeit von c_0 : Nach Hilfssatz 4.6 ist $F_{B_{n,p}}$ schwach monoton abnehmend in n . Mit 3. folgt deshalb

$$F_{B_{c_0, \Delta^+ p_i(t)}}(\mu_{l_i}^{-1}(\delta_l)) \geq F_{B_{c_0^i, \Delta^+ p_i(t)}}(\mu_{l_i}^{-1}(\delta_l)) \text{ für } i = 1, \dots, N, t = 0, 1, \dots, t_{\text{end}}. \quad (4.39)$$

Nach Hilfssatz 4.5 ist $F_{B_{n,p}}$ streng monoton abnehmend in p . Mit 1. folgt deshalb

$$F_{B_{c_0, \Delta^+ p_i(t)}}(\mu_{l_i}^{-1}(\delta_l)) \geq F_{B_{c_0^i, \Delta^+ p_i(t_{\max}^i)}}(\mu_{l_i}^{-1}(\delta_l)) \text{ für } i = 1, \dots, N, t = 0, 1, \dots, t_{\text{end}}. \quad (4.40)$$

Mit (4.35) folgt aus 2.:

$$F_{B_{c_0, \Delta^+ p_i(t_{\max}^i)}}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für } i = 1, \dots, N. \quad (4.41)$$

Damit folgt aus (4.39) – (4.41)

$$F_{B_{c_0, \Delta^+ p_i(t)}}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für } i = 1, \dots, N, t = 0, 1, \dots, t_{\text{end}},$$

das heißt c_0 ist eine zulässige Lösung.

Optimalität von c_0 : Der Beweis wird indirekt geführt. Sei c_0 nicht die maximale zulässige Lösung. Dann gibt es ein

$$c_0^* > c_0, \quad (4.42)$$

das eine zulässige Lösung ist, das heißt es besitzt die Eigenschaft

$$F_{B_{c_0^*, \Delta^+ p_i(t)}}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für alle } i = 1, \dots, N \text{ und für alle } t = 0, 1, \dots, t_{\text{end}},$$

also auch für $t = t_{\max}^i, i = 1, \dots, N$. Daher gilt

$$F_{B_{c_0^*, \Delta^+ p_i(t_{\max}^i)}}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für alle } i = 1, \dots, N. \quad (4.43)$$

Wegen (4.42) folgt aus 3., dass es ein c_0^i gibt, mit

$$c_0^i < c_0^*. \quad (4.44)$$

Wegen 2. gilt

$$c_0^i = \max \left\{ c_0' \mid F_{B_{c_0', \Delta^* p_1(t_{\max}^i)}}^{-1}(\delta_\zeta) = \mu_{1_i}^{-1}(\delta_1) \right\}, \quad (4.45)$$

wobei gemäß Def. 4.10 $F_{B_{c_0', \Delta^* p_1(t_{\max}^i)}}^{-1}(\delta_\zeta) = \inf \left\{ x : F_{B_{c_0', \Delta^* p_1(t_{\max}^i)}}(x) \geq \delta_\zeta \right\}$. Wegen (4.43) gehört auch c_0^* zur Menge der c_0' in (4.45). Daher ist $c_0^i \geq c_0^*$. Das steht im Widerspruch zu (4.44). Deshalb kann die Annahme (4.42) nicht zutreffen und c_0 ist die maximale zulässige Lösung. ■

4.2.3 Zweites Optimierungsproblem

Das zweite Optimierungsproblem lautet (vgl. Abschnitt 3.4):

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan innerhalb eines gegebenen Zeitraums unter Einhaltung der Kapazitätsschranken für Akquiseaktionen zugeführt werden kann, und wie ist sie über den Zeitraum verteilt?

Während beim ersten Optimierungsproblem nur zum Zeitpunkt $t = 0$ Kunden dem Akquiseplan $AP = (\underline{A}, \underline{R})$ zugeführt werden, wird jetzt angenommen, dass innerhalb des Zeitraums, der die Zeitpunkte

$$t = 0, 1, \dots, t_{\text{in}}$$

umfasst, Kunden dem Akquiseplan zugeführt werden können. Analog zum ersten Optimierungsproblem wird die Anzahl mit

$$c_\tau \in \mathbb{N}_0 \text{ für } \tau = 0, 1, \dots, t_{\text{in}}$$

bezeichnet. Die Anzahlen der zugeführten Kunden werden im Vektor $c^{t_{\text{in}}}$ zusammengefasst:

$$c^{t_{\text{in}}} = (c_0, c_1, \dots, c_{t_{\text{in}}}).$$

Wie beim ersten Optimierungsproblem wird davon ausgegangen, dass die Akquise endlich ist, das heißt es gibt einen Zeitpunkt t_{end} , zu dem sich alle Kunden entweder im Knoten A_{N-1} oder im Knoten A_N befinden. Daher wird fortan stillschweigend $t_{\text{in}} < t_{\text{end}}$ vorausgesetzt. Die Zielfunktion des zweiten Optimierungsproblems lautet

$$\sum_{\tau=0}^{t_{\text{in}}} c_\tau \stackrel{!}{=} \max.$$

Wenn zum Zeitpunkt $\tau \in \{0, 1, \dots, t_{\text{in}}\}$ die Anzahl c_τ Kunden dem Akquiseplan zugeführt wird, dann befinden sich von diesen Kunden zum Zeitpunkt $t \in \{\tau, \tau + 1, \dots, t_{\text{end}}\}$ im Knoten A_i , $i \in \{1, \dots, N\}$

$$\Delta^+\zeta(c_\tau, i, t - \tau), t \geq \tau \quad (4.46)$$

neue Kunden. Die an einem bestimmten Zeitpunkt dem Akquiseplan zugeführten Kunden verhalten sich genauso wie beim ersten Optimierungsproblem. Da sie aber nicht zum Zeitpunkt $t = 0$ zugeführt werden, sondern zum Zeitpunkt $t = \tau$, ist der Zeitparameter der Zufallsvariablen (4.46) $t - \tau$. Das wird deutlich, wenn für die Zeit t die Transformation

$$t \rightarrow t' \text{ mit } t' := t - \tau \text{ für } t \geq \tau$$

durchgeführt wird. Dann wird aus (4.46):

$$\Delta^+\zeta(c_0, i, t'), t' \geq 0. \quad (4.47)$$

Die Zufallsgröße $\Delta^+\zeta(c_0, i, t')$ folgt laut Satz 4.3 der Binomialverteilung $B_{c_0, \Delta^+p_i(t')}$, und deshalb gelten

$$\Delta^+\zeta(c_\tau, i, t - \tau) \sim B_{c_\tau, \Delta^+p_i(t - \tau)} \quad (4.48)$$

und damit

$$\mathbb{E}\Delta^+\zeta(c_\tau, i, t - \tau) = c_\tau \cdot \Delta^+p_i(t - \tau),$$

jeweils für $t \geq \tau$. Im Folgenden werden Größen eingeführt, die sich leichter definieren lassen, wenn $\Delta^+\zeta(c_\tau, i, t - \tau)$ auch für $t < \tau$ definiert ist. Das bedeutet, dass die Präsenzwahrscheinlichkeiten neuer Kunden, $\Delta^+p_i(t)$, $i = 1, \dots, N$ in Erweiterung von Def. 4.6 auch für negatives t zu definieren sind.

Definition 4.11: *Erweiterung von Def. 4.6.*

In Erweiterung von Def. 3.6 sei $p_i(t) := 0$ für $t \in \mathbb{Z}$, $t < 0$. □

Anmerkungen

1. Obige Definition kann auch aus Def. 3.4 und Def. 3.5 abgeleitet werden. Denn nach Def. 4.6 ist

$$\Delta^+p_j(t) := W\{A(j, t) \cap \bar{A}(j, t - 1)\}$$

die Wahrscheinlichkeit, dass zum Zeitpunkt t ein Kunde neu im Knoten A_j , $j = 1, \dots, N$ ist, wenn der Kunde zum Zeitpunkt $t = 0$ dem Akquiseplan zugeführt wurde. Das Ereignis $A(j, t)$ ist in Def. 3.5 über Knotenfolgen $(a_t) \in M$ definiert, die alle mit $t = 0$ beginnen:

$$A(j, t) := \bigcup_{(a_\tau)_{\tau \geq 0} \in M \mid a_t = j} \{(a_\tau)\}.$$

Somit gilt für das Ereignis $A(j, t) \in E$:

$$A(j, t) = \emptyset \text{ für } t \in \mathbb{Z}, t < 0. \quad (4.49)$$

Also kann sich, in Übereinstimmung mit der natürlichen Anschauung, kein Kunde in einem der Knoten befinden, bevor er dem Akquiseplan zugeführt worden ist. Mit (4.49) ist

$$\Delta^+ p_j(t) = W\{A(j, t) \cap \bar{A}(j, t-1)\} = W\{\emptyset \cap \bar{A}(j, t-1)\} = W\{\emptyset\}$$

und damit gemäß der Festlegungen für die Wahrscheinlichkeitsverteilung W in Def. 3.4:

$$\Delta^+ p_j(t) = W\{\emptyset\} = 0.$$

2. Nun sind die Zufallsgrößen $\Delta^+ \zeta(c_0, i, t)$ bzw. $\Delta^+ \zeta(c_\tau, i, t - \tau)$ auch für $t \in \mathbb{Z}, t < 0$ bzw. für $t \in \mathbb{N}_0, t < \tau$ definiert. Denn mit $\Delta^+ p_i(t) = 0$ bzw. $\Delta^+ p_i(t - \tau) = 0$ gilt für die Binomialverteilungen:

$$B_{c_0, \Delta^+ p_i(t)}(z) = \begin{cases} 1 & \text{für } z = 0 \\ 0 & \text{für } z \in \mathbb{N} \setminus \{0\} \end{cases} \quad \text{bzw.} \quad B_{c_\tau, \Delta^+ p_i(t-\tau)}(z) = \begin{cases} 1 & \text{für } z = 0 \\ 0 & \text{für } z \in \mathbb{N} \setminus \{0\} \end{cases},$$

jeweils für $i = 1, \dots, N$.

Die zu den Zeitpunkten $t = 0, 1, \dots, t_{\text{in}}$ dem Akquiseplan zugeführten Kunden verhalten sich unabhängig voneinander (vgl. dazu Anmerkung 2. zu Satz 4.3). Deshalb gilt das Superpositionsprinzip und die Zufallsgröße für das zweite Optimierungsproblem kann in Anlehnung an Def. 4.7 wie folgt formuliert werden:

Definition 4.12: *Zuführungszeitraum, Zufallsgröße der Anzahl neuer Kunden zum Zeitpunkt t im Knoten A_i .*

Sei $AP = (\underline{A}, \underline{R})$ ein Akquiseplan und bezeichne $t = 0$ den Beginn der Akquise, t_{in} den Zeitpunkt, zu dem dem Akquiseplan letztmals neue Kunden zugeführt werden und t_{end} das Akquiseende. Dann bezeichnet die Menge $\{0, 1, \dots, t_{\text{in}}\}$ den *Zuführungszeitraum* der Akquise und

$$\Delta^+ \zeta(c^{t_{\text{in}}}, i, t) = \sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} \Delta^+ \zeta(c_\tau, i, t - \tau)$$

die *Zufallsgröße der Anzahl neuer Kunden zum Zeitpunkt t im Knoten A_i* für die dem Akquiseplan zu den Zeitpunkten $\tau = 0, 1, \dots, t_{\text{in}}, t_{\text{in}} < t_{\text{end}}$ zugeführten

Kundenzahlen $c_\tau \in \mathbb{N}_0$, wobei $\Delta^+\zeta(c_\tau, i, t - \tau)$ eine Zufallsgröße gemäß Def. 4.7 ist, das heißt es gilt insbesondere

$$\Delta^+\zeta(c_\tau, i, t - \tau) \sim B_{c_\tau, \Delta^+p_i(t-\tau)},$$

wobei $\Delta^+p_i(t)$ die Präsenzwahrscheinlichkeit nach Def. 4.6 ist, die gemäß Satz 4.2 berechnet werden kann. \square

Anmerkungen

1. Diese Definition ist eine Verallgemeinerung von Def. 4.7, weil die Zufallsgröße $\Delta^+\zeta(c_0, i, t)$ nur ein Sonderfall von $\Delta^+\zeta(c^{t_{in}}, i, t)$ für $c_0 \in \mathbb{N}_0$ und $c_\tau = 0$ für $\tau = 1, \dots, t_{in}$ ist. Denn dann gilt

$$\Delta^+\zeta(c^{t_{in}}, i, t) = \sum_{\tau=0}^{\min\{t_{in}, t\}} \Delta^+\zeta(c_\tau, i, t - \tau) = \Delta^+\zeta(c_0, i, t) + \sum_{\tau=1}^{\min\{t_{in}, t\}} \Delta^+\zeta(0, i, t - \tau), \quad (4.50)$$

und da $\Delta^+\zeta(0, i, t) \equiv 0$ für alle $i = 1, \dots, N$ und alle $t = 0, 1, \dots, t_{end}$, bleibt von (4.50)

$$\Delta^+\zeta(c^{t_{in}}, i, t) = \Delta^+\zeta(c_0, i, t)$$

übrig.

2. Der Erwartungswert von $\Delta^+\zeta(c^{t_{in}}, i, t)$ für gegebenes i und festes t beträgt

$$\begin{aligned} \mathbb{E}\Delta^+\zeta(c^{t_{in}}, i, t) &= \left[\mathbb{E} \sum_{\tau=0}^{\min\{t_{in}, t\}} \Delta^+\zeta(c_\tau, i, t - \tau) \right] = \sum_{\tau=0}^{\min\{t_{in}, t\}} \mathbb{E}\Delta^+\zeta(c_\tau, i, t - \tau) \\ &= \sum_{\tau=0}^{\min\{t_{in}, t\}} c_\tau \Delta^+p_i(t - \tau), \end{aligned} \quad (4.51)$$

weil die Erwartungswertbildung ein lineares Funktional ist [155] (vgl. (4.26)).

3. Die Wahrscheinlichkeitsverteilung von $\Delta^+\zeta(c^{t_{in}}, i, t)$ für gegebenes i und festes t beträgt

$$\begin{aligned} W\{\Delta^+\zeta(c^{t_{in}}, i, t) = z\} &= W\left\{ \sum_{\tau=0}^{\min\{t_{in}, t\}} \Delta^+\zeta(c_\tau, i, t - \tau) = z \right\} \\ &= \sum_{(x_0, \dots, x_{\min(t_{in}, t)}) : \left(\sum_{\tau=0}^{\min(t_{in}, t)} x_\tau = z \right)} \left[\prod_{\tau=0}^{\min(t_{in}, t)} W\left\{ \Delta^+\zeta(c_\tau, i, t - \tau) = x_\tau \right\} \right]. \end{aligned} \quad (4.52)$$

Nach der Definition der Zufallsgröße kann das zweite Optimierungsproblem völlig analog zum ersten Optimierungsproblem formuliert werden. Es lautet dann:

$$\sum_{\tau=0}^{t_{\text{in}}} c_{\tau} \stackrel{!}{=} \max$$

unter den Nebenbedingungen $c_{\tau} \in \mathbb{N}_0$ für $\tau = 0, 1, \dots, t_{\text{in}}$ und

$$\Delta^+ \zeta(c^{t_{\text{in}}}, i, t) \leq \mu_{l_i}^{-1}(\delta_l) \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}. \quad (4.53)$$

Für die Behandlung der Ungleichungen (4.53) werden die gleichen Ansätze gewählt wie beim ersten Optimierungsproblem, im Abschnitt 4.2.3.1 die Formulierung als *Erwartungswertproblem* und im Abschnitt 4.2.3.2 den Ansatz der *Optimierung mit zufälligen Nebenbedingungen*.

4.2.3.1 Zweites Optimierungsproblem als Erwartungswertproblem

Das zweite Optimierungsproblem stellt sich als Erwartungswertproblem dar, wenn in (4.53) die Zufallsgröße $\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)$ durch ihren Erwartungswert $\mathbb{E} \Delta^+ \zeta(c^{t_{\text{in}}}, i, t)$ ersetzt wird, dessen Wert nach (4.51) gleich $\sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} c_{\tau} \Delta^+ p_i(t - \tau)$ ist. Damit lautet das Optimierungsproblem wie folgt:

Definition 4.13: *Zweites Optimierungsproblem als Erwartungswertproblem.*

Das zweite Optimierungsproblem zum Wahrheitsniveau δ lautet als *Erwartungswertproblem*:

$$\sum_{\tau=0}^{t_{\text{in}}} c_{\tau} \stackrel{!}{=} \max \quad (4.54)$$

unter den Nebenbedingungen $c_{\tau} \in \mathbb{N}_0$ für $\tau = 0, 1, \dots, t_{\text{in}}$ und

$$\sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} c_{\tau} \Delta^+ p_i(t - \tau) \leq \mu_{l_i}^{-1}(\delta_l) \quad (4.55)$$

für alle $i = 1, \dots, N$ und alle $t = 0, 1, \dots, t_{\text{end}}$. □

Anmerkungen

1. Das zweite Optimierungsproblem als Erwartungswertproblem ist ein ganzzahliges lineares Optimierungsproblem. Es kann mit Verfahren der ganzzahligen linearen Optimierung gelöst werden.
2. Der Lösungsraum $\mathbb{N}_0^{t_{\text{in}}}$ ist nicht leer. Er enthält unter anderem den Punkt $(c_0, \dots, c_{t_{\text{in}}})$ mit $c_1 = c_2 = \dots = c_{t_{\text{in}}} = 0$ und c_0 mit dem Wert (4.30) aus Satz 4.4.

3. Zur Problematik des Erwartungswertproblems wird auf die Anmerkungen zum Satz 4.4 verwiesen. Für die Varianz gilt [155]:

$$\begin{aligned} \text{Var } \Delta^+ \zeta(c^{t_{\text{in}}}, i, t) &= \sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} \text{Var } \Delta^+ \zeta(c_{\tau}, i, t-\tau) \\ &+ 2 \sum_{\tau=0}^{\min\{t_{\text{in}}, t\}-1} \left[\sum_{\vartheta=\tau+1}^{\min\{t_{\text{in}}, t\}} \text{Cov}(\Delta^+ \zeta(c_{\tau}, i, t-\tau), \Delta^+ \zeta(c_{\vartheta}, i, t-\vartheta)) \right] \end{aligned}$$

und $\text{Var } \Delta^+ \zeta(c_{\tau}, i, t-\tau) = c_{\tau} \Delta^+ p_i(t-\tau)[1 - \Delta^+ p_i(t-\tau)]$. Daraus ist ersichtlich, dass die Varianz von $\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)$ schwach monoton wachsend in den Optimierungsvariablen ist.

Die Alternative zum Erwartungswertproblem mit seinen Unzulänglichkeiten ist wie beim ersten Optimierungsproblem der Ansatz aus der *Optimierung mit zufälligen Nebenbedingungen*. Er wird sich hier aber als deutlich komplexer erweisen.

4.2.3.2 Optimierung mit zufälligen Nebenbedingungen

Bei der *Optimierung mit zufälligen Nebenbedingungen* wird eine Mindestwahrscheinlichkeit, das Wahrscheinlichkeitsniveau δ_{ζ} vorgegeben und gefordert, dass alle Nebenbedingungen in (4.53) mindestens mit der Wahrscheinlichkeit δ_{ζ} eingehalten werden, das heißt

$$\begin{aligned} W \left\{ \Delta^+ \zeta(c^{t_{\text{in}}}, i, t) \leq \mu_{l_i}^{-1}(\delta_l) \right\} &\geq \delta_{\zeta} \\ \text{für alle } i &= 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}. \end{aligned} \quad (4.56)$$

Mit Hilfe der Verteilungsfunktion von $\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)$,

$$F_{\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)}(x) := \sum_{z=0}^x W \{ \Delta^+ \zeta(c^{t_{\text{in}}}, i, t) = z \} \quad (4.57)$$

können die Nebenbedingungen (4.56) kürzer formuliert werden:

$$F_{\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_{\zeta} \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}}.$$

Im Gegensatz zum ersten Optimierungsproblem ist $F_{\Delta^+ \zeta(c^{t_{\text{in}}}, i, t)}$ sehr komplex:

$$\begin{aligned}
 F_{\Delta^+ \zeta(\mathbf{c}^{t_{\text{in}}}, \mathbf{i}, t)}(\mathbf{x}) &:= \sum_{z=0}^x W\{\Delta^+ \zeta(\mathbf{c}^{t_{\text{in}}}, \mathbf{i}, t) = z\} \\
 &= \sum_{z=0}^x \sum_{(x_0, \dots, x_{\min(t_{\text{in}}, t)}) : \sum_{\tau=0}^{\min(t_{\text{in}}, t)} x_\tau = z} \left[\prod_{\tau=0}^{\min(t_{\text{in}}, t)} W\left\{\Delta^+ \zeta(\mathbf{c}_\tau, \mathbf{i}, t - \tau) = x_\tau\right\} \right] \\
 &= \sum_{z=0}^x \sum_{(x_0, \dots, x_{\min(t_{\text{in}}, t)}) : \sum_{\tau=0}^{\min(t_{\text{in}}, t)} x_\tau = z} \left[\prod_{\tau=0}^{\min(t_{\text{in}}, t)} \left[\binom{\mathbf{c}_\tau}{\mathbf{x}_\tau} (\Delta^+ \mathbf{p}_i(t - \tau))^{x_\tau} (1 - \Delta^+ \mathbf{p}_i(t - \tau))^{c_\tau - x_\tau} \right] \right].
 \end{aligned}$$

Damit lauten die Nebenbedingungen (4.56):

$$\sum_{z=0}^{\mu_i^{-1}(\delta_i)} \sum_{(x_0, \dots, x_{\min(t_{\text{in}}, t)}) : \sum_{\tau=0}^{\min(t_{\text{in}}, t)} x_\tau = z} \left[\prod_{\tau=0}^{\min(t_{\text{in}}, t)} \left[\binom{\mathbf{c}_\tau}{\mathbf{x}_\tau} (\Delta^+ \mathbf{p}_i(t - \tau))^{x_\tau} (1 - \Delta^+ \mathbf{p}_i(t - \tau))^{c_\tau - x_\tau} \right] \right] \geq \delta_\zeta$$

für alle $i = 1, \dots, N$ und alle $t = 0, 1, \dots, t_{\text{end}}$.

Die folgende Definition fasst obige Ausführungen zusammen.

Definition 4.14: *Zweites Optimierungsproblem als Optimierungsproblem mit zufälligen Nebenbedingungen.*

Das zweite Optimierungsproblem zum Wahrheitsniveau δ_i und zum Wahrscheinlichkeitsniveau δ_ζ lautet als *Optimierungsproblem mit zufälligen Nebenbedingungen*:

$$\sum_{\tau=0}^{t_{\text{in}}} c_\tau \stackrel{!}{=} \max \quad (4.58)$$

unter den Nebenbedingungen $c_\tau \in \mathbb{N}_0$ für $\tau = 0, 1, \dots, t_{\text{in}}$ und

$$\sum_{z=0}^{\mu_i^{-1}(\delta_i)} \sum_{(x_0, \dots, x_{\min(t_{\text{in}}, t)}) : \sum_{\tau=0}^{\min(t_{\text{in}}, t)} x_\tau = z} \left[\prod_{\tau=0}^{\min(t_{\text{in}}, t)} \left[\binom{\mathbf{c}_\tau}{\mathbf{x}_\tau} (\Delta^+ \mathbf{p}_i(t - \tau))^{x_\tau} (1 - \Delta^+ \mathbf{p}_i(t - \tau))^{c_\tau - x_\tau} \right] \right] \geq \delta_\zeta$$

für alle $i = 1, \dots, N$ und alle $t = 0, 1, \dots, t_{\text{end}}$. (4.59)

□

Anmerkungen

1. Die Zielfunktion (4.58) ist linear.

2. Die Nebenbedingungen (4.59) sind in den Optimierungsvariablen $c_0, \dots, c_{t_{in}}$ nichtlinear. Damit ist das zweite Optimierungsproblem als *Optimierungsproblem mit zufälligen Nebenbedingungen* ist ein *nichtlineares* ganzzahliges Optimierungsproblem.

Im Folgenden wird untersucht, ob das Problem gemäß Def. 4.14 grundsätzlich lösbar ist und wie es gelöst werden könnte. Zunächst sei ein allgemeiner Ansatz zur Lösung ganzzahliger Optimierungsprobleme betrachtet.

Gegeben sei ein ganzzahliges Optimierungsproblem³⁾,

$$\max_x (cx) \text{ für } x \in S \subseteq \mathbb{N}_0^n. \quad (4.60)$$

Es gebe ein Polyeder, $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$, so dass für die Menge der zulässigen Punkte von (4.60) gilt:

$$S = P \cap \mathbb{Z}^n. \quad (4.61)$$

Mit der Bezeichnung $\text{conv}(S)$ für die konvexe Hülle⁴⁾ von S kann folgendes ganzzahliges Optimierungsproblem formuliert werden:

$$\max_x (cx) \text{ für } x \in \text{conv}(S). \quad (4.62)$$

Über die Optimierungsprobleme (4.60) und (4.62) macht [120] folgende Aussagen:

- a) Wenn die Zielfunktion des Optimierungsproblems (4.62) in $\text{conv}(S)$ beschränkt ist, dann besitzt das Problem eine optimale Lösung x^0 , die sich auf dem Rand von $\text{conv}(S)$ befindet, und die eine optimale Lösung des Optimierungsproblems (4.60) ist.
- b) Wenn x^0 eine optimale Lösung von (4.60) ist, dann ist x^0 auch eine optimale Lösung von (4.62).

Der Satz sagt aus, dass das ganzzahlige Problem (4.60) durch Lösen des reellwertigen Problems (4.62) gelöst wird, vorausgesetzt, S liegt in einem Polyeder des \mathbb{R}^n . In diesem Fall ist eine Lösung einfach berechenbar, wenn aus der Darstellung von P als Menge linearer Ungleichungen eine ebensolche Darstellung für $\text{conv}(S)$ abgeleitet werden kann. In obiger Formulierung des zweiten Optimierungsproblems ist die Menge der zulässigen Punkte S durch die Nebenbedingungen (4.59) gegeben. Die Schwierigkeit besteht in der Herleitung eines Poly-

3. Im Gegensatz zum ersten und zweiten Optimierungsproblem bezeichnet hier c den Vektor der Zielfunktionskoeffizienten.

4. Die konvexe Hülle von $S \subseteq \mathbb{R}^n$ ist die Menge aller Linearkombinationen von Punkten aus S :

$$\text{conv}(S) := \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^m \lambda_i x^i, \sum_{i=1}^m \lambda_i = 1, x^i \in S \text{ für } i = 1, \dots, m, m \leq |S| \right\}.$$

ders P mit der Eigenschaft (4.61) sowie in der Beschreibung der konvexen Hülle in (4.62). Dieser Weg wird hier nicht weiter verfolgt. Im Abschnitt 4.3 wird ein alternatives, szenarien-basiertes Modell des Problems vorgestellt, das sich einfacher lösen lässt.

Über die grundsätzliche Lösbarkeit des zweiten Optimierungsproblems gibt der folgende Satz Auskunft:

Satz 4.8: Für das zweite Optimierungsproblem als *Optimierungsproblem mit zufälligen Nebenbedingungen* gemäß Def. 4.14 existiert eine Lösung.

□

Beweis: Es wird angenommen, dass die Menge der zulässigen Punkte des Optimierungsproblems endlich ist: Sei $C \subset \mathbb{N}_0^{t_{\text{in}}+1}$ eine endliche Obermenge der Menge zulässiger Punkte des Optimierungsproblems, das heißt

$$C = \{c^{t_{\text{in}},1}, \dots, c^{t_{\text{in}},r}\} \text{ mit } r = |C| \quad (4.63)$$

und $c^{t_{\text{in}},j} = (c_0^j, c_1^j, \dots, c_{t_{\text{in}}}^j)^T$. Dann löst Alg. 4.1 das Optimierungsproblem gemäß Def. 4.14. Unterprogramme zur Berechnung der Zulässigkeit eines Elements aus C sind in Alg. 4.3 und Alg. 4.4 dargestellt. Jetzt ist noch zu zeigen, dass eine endliche Obermenge C existiert. Seien

$$\tilde{c}^\tau := (c_0^\tau, c_1^\tau, \dots, c_{t_{\text{in}}}^\tau)^T \text{ mit } c_i^\tau := \delta_{i\tau} c_\tau^{\text{opt}} \quad (4.64)$$

für $\tau = 0, \dots, t_{\text{end}}$, wobei $\delta_{i\tau}$ das so genannte *Kronecker-Symbol* ist⁵⁾, und c_τ^{opt} ist die Lösung des ersten Optimierungsproblems gemäß Def. 4.9, wenn die Akquise mit $t = \tau$ beginnt und mit $t = t_{\text{end}}$ endet. Die Lösung kann gemäß Satz 4.7 berechnet werden, indem t durch $t - \tau$ substituiert wird. Offensichtlich gehören

$$\begin{aligned} \tilde{c}^0 &:= (c_0^{\text{opt}}, 0, \dots, 0)^T, \\ \tilde{c}^1 &:= (0, c_1^{\text{opt}}, 0, \dots, 0)^T, \\ \tilde{c}^2 &:= (0, 0, c_2^{\text{opt}}, 0, \dots, 0)^T, \\ &\dots \\ \tilde{c}^{t_{\text{in}}} &:= (0, \dots, 0, c_{t_{\text{in}}}^{\text{opt}})^T \end{aligned}$$

zur Menge C . Die c_τ^{opt} , $\tau = 0, 1, \dots, t_{\text{in}}$ sind optimal, also insbesondere so groß, dass jeder zusätzliche Kunde im Akquiseplan zur Verletzung einer der Kapazitätsschranken führen würde. Deshalb muss der Vektor

5. Für alle $i, j \in \mathbb{N}$ gilt: $\delta_{ij} = 1$ für $i = j$ und $\delta_{ij} = 0$ für $i \neq j$ [71].

$$(c_0^{\text{opt}}, c_1^{\text{opt}}, c_2^{\text{opt}}, \dots, c_{t_{\text{in}}}^{\text{opt}})^T \quad (4.65)$$

eine obere Schranke für die Lösung des zweiten Optimierungsproblems gemäß Def. 4.14 sein, und

$$C := \left\{ (c_0, c_1, c_2, \dots, c_{t_{\text{in}}})^T \mid \forall \tau = 0, 1, \dots, t_{\text{in}} : c_\tau \in \{0, 1, \dots, c_\tau^{\text{opt}}\} \right\}$$

ist damit eine endliche Obermenge der Menge der zulässigen Punkte mit

$$r = \prod_{\tau=0}^{t_{\text{in}}} (c_\tau^{\text{opt}} + 1) \quad (4.66)$$

Elementen. ■

```

Führe aus
  BerechneDimC(r);                                // Glg. 4.63
Eingabe:  $\delta_l, \delta_c$ ;                               // Wahrheitsniveau,
                                                    // Wahrscheinlichkeitsniveau

Für  $i = 2, \dots, N$  führe aus
   $u := \mu_i^{-1}(\delta_l)$ ;                               // Def. 4.3

Setze  $\text{zul} := \text{'falsch'}$ ,  $j := 0$ ;
Solange  $\text{zul} = \text{'falsch'}$  führe aus {
  Setze  $j := j + 1$ ;
   $\text{zul} := \text{IstZulässig}(c^{t_{\text{in}},j})$ ;                // Unterprog. Alg. 4.3
}

Setze  $j_{\text{anf}} := j$ ,  $c^{\text{opt}} := c^{t_{\text{in}},j_{\text{anf}}}$ ;
Für  $j = j_{\text{anf}} + 1, \dots, r$  führe aus {
  Wenn  $\text{IstZulässig}(c^{t_{\text{in}},j}) = \text{'wahr'}$  dann
    Wenn  $\text{Summe}(c^{t_{\text{in}},j}) > \text{Summe}(c^{\text{opt}})$  dann // Unterprog. Alg. 4.4
      Setze  $c^{\text{opt}} := c^{t_{\text{in}},j}$ ;
  }
Ausgabe:  $c^{\text{opt}}$ ;                                // löst das Problem gemäß Def. 4.14

```

Algorithmus 4.1: Lösung des Optimierungsproblems gemäß Def. 4.14 durch vollständige Aufzählung.

Die im Beweis von Satz 4.8 gezeigte Möglichkeit der Lösung des zweiten Optimierungsproblems als *Optimierungsproblem mit zufälligen Nebenbedingungen* durch vollständige Aufzählung ist unbefriedigend, weil sie mit einem sehr großen Rechenaufwand verbunden ist. Dazu gehören:

1. Die t_{in} -fache Berechnung von Lösungen des ersten Optimierungsproblems als *Optimierungsproblem mit zufälligen Nebenbedingungen* (vgl. Alg. 4.2 und Def. 4.9).
2. Die große Anzahl der Elemente der Menge C (vgl. (4.66)).


```

Definiere BerechneDimC(r) {                                     // Berechnung von Glg. 4.66
    Setze  $r := 1$ ,  $C := (0, \dots, 0)^T$ ;
    Für  $\tau := 0, \dots, t_{in}$  führe aus {
        Setze  $t' := t - \tau$ ;
        Löse das Optimierungsproblem gemäß Def. 4.9 für  $t'$ ;
        Setze  $C := C + (0, \dots, c_{\tau}^{opt}, \dots, 0)$ ;           // vgl. Glg. 4.65
        Setze  $r := r \cdot c_{\tau}^{opt}$ ;
    }
    Ausgabe:  $r$ ,  $C$ ;
}
    
```

Algorithmus 4.2: Unterprogramm *BerechneDimC(r)*.

```

Definiere IstZulässig(ctin) {
    Setze  $zul := \text{'falsch'}$ ,  $j := 0$ ;
    Solange  $zul = \text{'falsch'}$  führe aus {
        Setze  $weiter := \text{'wahr'}$ ,  $i := 1$ ;
        Solange  $weiter = \text{'wahr'}$  und  $i < N$  führe aus {
            Setze  $i := i + 1$ ,  $t := 0$ ;
            Solange  $weiter = \text{'wahr'}$  und  $t < t_{end}$  führe aus {
                Setze  $t := t + 1$ ;
                Berechne  $F_{\Delta^+ \zeta(c^{tin,i}, i, t)}(\mu_{l_i}^{-1}(\delta_l))$ ;           // Glg. 4.52
                Wenn  $F_{\Delta^+ \zeta(c^{tin,i}, i, t)}(\mu_{l_i}^{-1}(\delta_l)) < \delta_{\zeta}$  dann
                    Setze  $weiter := \text{'falsch'}$ ;
            }
        }
        Wenn  $weiter = \text{'wahr'}$  dann setze  $zul := \text{'wahr'}$ ;
    }
    Ausgabe:  $zul$ ;
}
    
```

Algorithmus 4.3: Unterprogramm *IstZulässig(c^{tin})*.

```

Definiere Summe(ctin) {                                     // Berechnung von Glg. 4.58
    Setze  $summe := 0$ ;
    Für  $\tau := 0, \dots, t_{end}$  führe aus
         $summe := summe + c_{\tau}^{tin}$ ;
    Ausgabe:  $summe$ ;
}
    
```

Algorithmus 4.4: Unterprogramm *Summe(c^{tin})*.

3. Der hohe Rechenaufwand für die Überprüfung der Elemente der Menge C auf Zulässigkeit (vgl. (4.52)). Dieser Aufwand wird zwar durch geschickte Abbruchbedingungen verringert werden (vgl. Alg. 4.3), die Glg. 4.52 ist aber mindestens einmal für jedes Element in C zu berechnen.

Im Abschnitt 4.3 wird ein szenarienbasiertes Modell des zweiten Optimierungsproblems vorgestellt, das sich einfacher lösen lässt.

4.2.4 Drittes Optimierungsproblem

Das dritte Optimierungsproblem lautet (vgl. Abschnitt 3.4):

Wie muss eine vorgegebene Anzahl von Kunden einem Akquiseplan unter Einhaltung der Kapazitätsschranken zugeführt werden, damit der dafür benötigte Zeitraum möglichst kurz ist?

Für die Einhaltung der Kapazitätsschranken werden hier, wie beim ersten und beim zweiten Optimierungsproblem zwei Ansätze betrachtet: die Forderung, dass die Erwartungswerte der Anzahlen neuer Kunden in den Knoten des Akquiseplans den Kapazitätsschranken genügen und die Forderung, dass die Kapazitätsschranken nur mit einer vorgegebenen Wahrscheinlichkeit verletzt werden dürfen. Entsprechend fällt die Formulierung des dritten Optimierungsproblems aus.

Definition 4.15: *Drittes Optimierungsproblem als Erwartungswertproblem / als Optimierungsproblem mit zufälligen Nebenbedingungen.*

Für eine vorgegebene Kapazität $c \in \mathbb{N}_0$ lautet das *dritte Optimierungsproblem* zum Wahrheitsniveau δ :

$$t_{\text{in}} \stackrel{!}{=} \min$$

unter folgenden Nebenbedingungen:

1. als *Erwartungswertproblem*:

$$t_{\text{in}} \in \mathbb{N}_0 \text{ und}$$

$$\sum_{\tau=0}^{t_{\text{in}}} c_{\tau} \geq c \text{ und}$$

$$c_{\tau} \in \mathbb{N}_0 \text{ für } \tau = 0, 1, \dots, t_{\text{in}} \text{ und}$$

$$\sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} c_{\tau} \Delta^+ p_i(t-\tau) \leq \mu_{l_i}^{-1}(\delta_l) \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}};$$

2. als *Optimierungsproblem mit zufälligen Nebenbedingungen* zum Wahrscheinlichkeitsniveau δ_{ζ} :

$$t_{\text{in}} \in \mathbb{N}_0 \text{ und}$$

$$\sum_{\tau=0}^{t_{\text{in}}} c_{\tau} \geq c \text{ und}$$

$c_\tau \in \mathbb{N}_0$ für $\tau = 0, 1, \dots, t_{\text{in}}$ und

$$F_{\Delta^+\zeta(c^{t_{\text{in}}, i, t})}(\mu_{l_i}^{-1}(\delta_l)) \geq \delta_\zeta \text{ für alle } i = 1, \dots, N \text{ und alle } t = 0, 1, \dots, t_{\text{end}},$$

wobei $F_{\Delta^+\zeta(c^{t_{\text{in}}, i, t})}$ die Verteilungsfunktion (4.52) für gegebenes $i \in \{1, \dots, N\}$ und festes $t = 0, 1, \dots, t_{\text{end}}$ ist.

□

Das dritte Optimierungsproblem ist ein ganzzahliges nichtlineares Optimierungsproblem. Mit der Intervallschachtelung in Alg. 4.5 wird es auf die mehrfache Lösung des zweiten Optimierungsproblems zurückgeführt. Wird das zweite Optimierungsproblem als Erwartungswertproblem gelöst, dann löst Alg. 4.5 das dritte Optimierungsproblem ebenfalls als Erwartungswertproblem, wenn das zweite Optimierungsproblem als *Optimierungsproblem mit zufälligen Nebenbedingungen* gelöst wird, dann löst Alg. 4.5 auch das dritte Optimierungsproblem als *Optimierungsproblem mit zufälligen Nebenbedingungen*. Für die Durchführbarkeit von Alg. 4.5 ist es nur erforderlich, das zweite Optimierungsproblem zu lösen – unabhängig vom Lösungsansatz.

4.3 Unscharfes stochastisches Mehrgüterfluss-Problem

Mit den beiden Ansätzen zur Lösung des Problems der optimalen Beschickung eines Akquise-Plans aus Abschnitt 4.2 wurden mit dem *Erwartungswertansatz* und der *Optimierung mit zufälligen Nebenbedingungen* zwei Möglichkeiten zur Berechnung eines deterministischen Äquivalents vorgestellt. Trotz ihrer Geradlinigkeit ist die Eignung der Verfahren diskussionswürdig.

- Der Erwartungswertansatz (siehe Def. 4.13) ist ein gemischt-ganzzahliges lineares Optimierungsproblem. Angesichts gebrochener Koeffizienten in den Nebenbedingungen ist nicht anzunehmen, dass nach Relaxation der Ganzzahligkeitsforderung das reellwertige Problem eine ganzzahlige Lösung besitzt. Runden der gebrochenen Lösung ist nur unter bestimmten Voraussetzungen zu rechtfertigen. Die Verwendung eines speziellen Lösungsverfahrens für gemischt-ganzzahlige Optimierungsprobleme kann zu sehr langen Laufzeiten führen.
- Bei der *Optimierung mit zufälligen Nebenbedingungen* (siehe Def. 4.14) kommt zur Ganzzahligkeitsforderung erschwerend die Nichtlinearität der Nebenbedingungen hinzu.

Beide Ansätze berücksichtigen eventuell vorhandene Struktureigenschaften des Optimierungsproblems nicht. In diesem Abschnitt wird gezeigt, dass solche Eigenschaften vorhanden sind. Tatsächlich ist das zweite Optimierungsproblem ein Mehrgüterfluss-Problem mit „unbequemen“ Nebenbedingungen. Es unterscheidet sich von den Ansätzen des vorigen Abschnitts auch durch seine Zufallsvariablen. Sie werden im Abschnitt 4.3.2 eingeführt und dann werden im Abschnitt 4.3.3 die beiden Optimierungsprobleme mit diesen Zufallsvariablen beschrieben. Danach arbeitet der Abschnitt 4.3.4 das Mehrgüterfluss-Problem heraus und diskutiert seine Ganzzahligkeitseigenschaften. In diesem Zusammenhang werden insbesondere die beiden Arten von „unbequemen“ Nebenbedingungen untersucht. Im Abschnitt 4.3.4.3 wird gezeigt, dass die Ganzzahligkeit der Lösung trotz Bündel-Nebenbedingungen erhalten bleibt. Auf die

```

Eingabe:  $c \in \mathbb{N}_0$  ;
Setze  $k := 1$  ,  $t_{us}^1 := 1$  ,  $t_{os}^1 := t_{end}$  ,  $stopp := \text{'falsch'}$  ;
Solange  $stopp = \text{'falsch'}$  führe aus {
    Wähle  $t^k \in \{t_{us}^1 + 1, \dots, t_{os}^1 - 1\}$  ;           // zufällige Auswahl
                                                    // oder z. B.  $t^k := \left\lfloor \frac{t_{us}^1 + t_{os}^1}{2} \right\rfloor$ 

    LöseOP2( $t^k, (c_0^k, c_1^k, \dots, c_k^k)^T$ );           // Unterprogramm zur Lösung des 2.
                                                    // Optimierungsproblems mit  $t_{in} := t^k$ 

    Setze  $c^{t^k} := \sum_{\tau=0}^k c_{\tau}^{t^k}$  ;

    Wenn  $c^{t^k} < c$  dann führe aus {
        Setze  $t_{us}^{k+1} := t^k$  ;
        Setze  $t_{os}^{k+1} := t_{os}^k$  ;
    }
    Sonst führe aus {
        Setze  $t_{us}^{k+1} := t_{us}^k$  ;
        Setze  $t_{os}^{k+1} := t^k$  ;
    }
    Wenn  $t_{us}^{k+1} - t_{os}^{k+1} = 1$  dann
        Setze  $stopp := \text{'wahr'}$  ,  $k := k + 1$  ;
    }

    LöseOP2( $t_{us}^k, (c_0^k, c_1^k, \dots, c_{t_{us}^k}^k)^T$ );           // Unterprogramm zur Lösung des 2.
                                                    // Optimierungsproblems mit  $t_{in} := t_{us}^k$ 

    LöseOP2( $t_{os}^k, (c_0^k, c_1^k, \dots, c_{t_{os}^k}^k)^T$ );           // bzw.  $t_{in} := t_{os}^k$ 

    Setze  $c_{us}^{t_{us}^k} := \sum_{\tau=0}^{t_{us}^k} c_{\tau}^{t_{us}^k}$  ,  $c_{os}^{t_{os}^k} := \sum_{\tau=0}^{t_{os}^k} c_{\tau}^{t_{os}^k}$  ;

    Wenn  $c < c_{us}^{t_{us}^k}$  dann setze  $t_{in} := t_{us}^k$  ;
    Sonst setze  $t_{in} := t_{os}^k$  ;
Ausgabe:  $t_{in}$  ;

```

Algorithmus 4.5: Lösung des dritten Optimierungsproblems.

Behandlung der stochastischen Nebenbedingungen gehen die Abschnitte 4.3.4.4 – 4.3.4.5 ein. Diese Nebenbedingungen bleiben entweder als Nebenbedingungen erhalten oder sie werden mit Hilfe der Lagrange-Relaxation in die Zielfunktion „verschoben“. Das deterministische Äquivalent wird in beiden Fällen auf Grundlage von Szenarien erzeugt (Abschnitt 4.3.4.4). Zum Schluss wird im Abschnitt 4.4 ein Fallbeispiel betrachtet.

4.3.1 Modellierungsgrundlagen

In Def. 3.1 wird ein Akquiseplan AP als gerichteter Graph mit der Knotenmenge \underline{A} und der Kantenmenge \underline{R} eingeführt, der genau eine Quelle, zwei Senken und bestimmte Eigenschaften besitzt. Die kompakte Beschreibung eines Akquiseplans AP gibt seine Adjazenzmatrix A_{AP} in Def. 3.2. Mit ihrer Hilfe werden in Def. 3.13 für jeden Knoten des Akquiseplans die Menge der Vorgängerknoten und die Menge der Nachfolgerknoten beschrieben, die in folgender Definition benötigt werden.

Definition 4.16: *Minimaler Abstand eines Knotens von der Quelle, minimaler Abstand eines Knotens von den Senken.*

Sei $AP = (\underline{A}, \underline{R})$ ein Akquiseplan mit der Quelle A_q und den beiden Senken A_p und A_n . Der *minimale Abstand eines Knotens von der Quelle*,

$$d_q(A_i)$$

ist für $A_i \in \underline{A}$ wie folgt rekursiv definiert:

$$d_q(A_q) := 0$$

und

$$d_q(A_j) := 1 + \min_{i \in \text{Vorg}(j)} d_q(A_i) \text{ für } A_j \in \underline{A}.$$

Der *minimale Abstand eines Knotens von den Senken*,

$$d_s(A_i) \text{ für } A_i \in \underline{A}$$

ist das Maximum der minimalen Abstände des Knotens von den beiden Senken:

$$d_s(A_i) := \max \{ d_p(A_i), d_n(A_i) \},$$

die wie folgt rekursiv definiert sind:

$$d_p(A_p) = d_n(A_n) := 0$$

und

$$d_p(A_i) := 1 + \min_{j \in \text{Nachf}(i)} d_p(A_j) \text{ und } d_n(A_i) := 1 + \min_{j \in \text{Nachf}(i)} d_n(A_j) \text{ für } A_i \in \underline{A}.$$

□

4.3.2 Zufallsvariable

Gegeben sei ein Akquiseplan AP mit den Knoten A_i , $i \in I$, $I := \{1, \dots, N\}$ und der Adjazenzmatrix $A = (a_{ij})$. Die Knoten seien wieder so indiziert, dass $a_{ij} = 0$ für $j \leq i$. Betrachte

tet werde eine Akquise, die zum Zeitpunkt $t = 0$ beginnt und zum endlichen Zeitpunkt $t = t_{\text{end}}$ endet.

Für die Formulierung des Optimierungsproblems im nächsten Abschnitt wird eine Zufallsvariable benötigt, die den Anteil der Kunden bezeichnet, der im Zeitintervall $(t-1, t]$, $t = 1, \dots, t_{\text{end}}$ vom Knoten A_i zum Knoten A_j wechselt, $i, j \in I$ mit $a_{ij} = 1$. Zunächst wird der absolute Anteil der Kunden bezeichnet, der diesen Wechsel vollzieht.

Sei $\zeta_{ij}(t)$ die Zufallsvariable der Anzahl der Kunden, die im Zeitintervall $(t-1, t]$ vom Knoten A_i zum Knoten A_j wechselt, $i, j \in I$ mit $a_{ij} = 1$. Für $i = j$ wird $\zeta_{ii}(t)$ definiert als die Anzahl der Kunden, die im Knoten A_i bleiben. (Das ist die Anzahl der Kunden zum Zeitpunkt $t-1$ im Knoten A_i abzüglich aller Kunden, die im Zeitintervall $(t-1, t]$ zu einem anderen Knoten wechseln.) Wenn sich genau ein Kunde im Akquiseplan befindet, dann ist aus Satz 3.12 im Abschnitt 3.2 bekannt, dass die Zufallsvariable $\zeta_{ij}(t)$ für feste i, j und t einer Alternativverteilung folgt:

$$\zeta_{ij}(t) \sim A_{\Psi_{ij}^t}.$$

Bei $c > 1$ Kunden im Akquiseplan würde der Vektor $(\zeta_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ für feste i und t einer Multinomialverteilung über die Knoten A_j , $j \in \overline{\text{Nachf}(i)}$ folgen:

$$(\zeta_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} \sim M_{c, (\Psi_{ij}^t)_{j \in \overline{\text{Nachf}(i)}}} \quad (4.67)$$

[155]. Wenn im Knoten A_i zum Zeitpunkt $t-1$ genau $\gamma(t-1)$ Kunden vorhanden wären (wobei $\gamma(t-1) \leq c$ für alle $t = 1, \dots, t_{\text{end}}$), dann wäre

$$\frac{\zeta_{ij}(t)}{\gamma(t-1)} \quad (4.68)$$

eine Zufallsvariable für den relativen Anteil der Kunden, die den Knotenwechsel von A_i nach A_j vollziehen. Mit dieser Darstellung wäre die Zufallsvariable wieder abhängig von der Anzahl der Kunden im Akquiseplan und somit abhängig von den Optimierungsvariablen. Gerade diese Abhängigkeit soll aber vermieden werden, weil sie im Abschnitt 4.2.3 zu den dort genannten Problemen geführt hat. Deshalb wird eine Zufallsvariable,

$$\xi_{ij}(t) := \frac{\zeta_{ij}(t)}{c} \quad (4.69)$$

für $\zeta_{ij}(t)$ wie in (4.67) und ein festes, großes $c \in \mathbb{N}$, festgelegt. Entsprechend der oben genannten Festlegung für $\zeta_{ii}(t)$ gilt für $\xi_{ii}(t)$, $i \in I$:

$$\xi_{ii}(t) := 1 - \sum_{j \in \text{Nachf}(i)} \xi_{ij}.$$

Die Zufallsvariablen $\xi_{ij}(t)$ werden für gegebenes i und $j \in \overline{\text{Nachf}(i)}$ entsprechend (4.67) in einem Vektor $(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ zusammengefasst. Zur Berechnung der Wahrscheinlichkeiten seiner Ausprägungen werden die Mengen C und Q_c^n eingeführt:

$$C := \{0, 1, \dots, c\}, N_c^n := \left\{ k \in C^n \mid \sum_{j=1}^n k_j = c \right\}, Q_c^n := \left\{ q \in Q^n \mid q = \frac{1}{c} k, k \in N_c^n \right\}.$$

Dann ist mit (4.69):

$$W \left\{ (\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} = q, q \in Q_c^{|\overline{\text{Nachf}(i)}|} \right\} = W \left\{ (\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} = k, k \in N_c^n \right\},$$

und wegen der Multinomialverteilung von $(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ folgt auch $(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ einer Multinomialverteilung, so dass mit $q_j = k_j/c$ gilt:

$$W \left\{ (\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} = q, q \in Q_c^{|\overline{\text{Nachf}(i)}|} \right\} = \frac{c!}{\prod_{j \in \overline{\text{Nachf}(i)}} (k_j!)} \cdot \prod_{j \in \overline{\text{Nachf}(i)}} (\Psi_{ij}^t)^{k_j}.$$

Nach dieser Festlegung stellt sich die Frage, wie c zu wählen ist. Da (4.69) an (4.68) angelehnt ist, das heißt c der Anzahl der Kunden im betrachteten Knoten entspricht, sollte c so groß wie die an einem Knoten im Verlauf der Akquise höchstens vorkommende Anzahl von Kunden sein. Da die dem Akquiseplan zugeführte Anzahl von Kunden die Lösung des Optimierungsproblems ist, kann c nur ungefähr geschätzt werden.

Die Besonderheiten der Wahrscheinlichkeitsverteilung von $(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ hält folgende Definition fest.

Definition 4.17: *Normierte Multinomialverteilung.*

Ein Zufallsvektor $(\xi_j)_{j \in J}$, der einer *normierten Multinomialverteilung* mit den Parametern $c \in \mathbb{N}$ und $p_j \in [0, 1]$, $\sum_{j \in J} p_j = 1$ gehorcht, besitzt die Punktwahrscheinlichkeiten

$$W \{ (\xi_j)_{j \in J} = q \} = \begin{cases} \frac{c!}{\prod_{j \in J} (k_j!)} \cdot \prod_{j \in J} (p_j)^{k_j} & \text{für } q = \frac{1}{c} k \in Q_c^{|J|} \\ 0 & \text{sonst} \end{cases}.$$

Diese normierte Multinomialverteilung wird mit

$$M_{c, (p_j)_{j \in J}}^n$$

bezeichnet, also

$$(\xi_j)_{j \in J} \sim M_{c, (p_j)_{j \in J}}^n.$$

□

Jetzt kann der stochastische Prozess der Kundenverteilung eingeführt werden:

Definition 4.18: *Kundenverteilung, stochastischer Prozess der Kundenverteilung, Szenario der Kundenverteilung, Stichprobennahme der Kundenverteilung, Stichprobe der Kundenverteilung.*

Gegeben sei ein Akquiseplan AP mit den Knoten A_1, A_2, \dots, A_N , dem Akquisebeginn $t = 0$, dem Akquiseende $t = t_{\text{end}}$ und der Zufallsmatrix

$$\Xi(t) := (\xi_{ij}(t))_{i,j} \in [0, 1]^{N \times N},$$

die *Kundenverteilung*⁶⁾ genannt wird, für $t = 0, 1, \dots, t_{\text{end}}$ und mit

$$(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} \sim M_{c, (\Psi_{ij}^t)_{j \in \overline{\text{Nachf}(i)}}}^n, \quad (4.70)$$

wobei $c \in \mathbb{N}$ eine festgelegte Konstante ist und Ψ_{ij}^t die Übergangswahrscheinlichkeit vom Knoten A_i zum Knoten A_j im Zeitraum $(t-1, t]$, die gemäß Satz 3.12 berechnet werden. Dann bezeichnet die Folge $(\Xi(t))_{t=0}^{t_{\text{end}}}$ den *stochastischen Prozess der Kundenverteilung*. Eine Realisierung der Folge wird *Szenario der Kundenverteilung* genannt und mit $(X(t))_{t=0}^{t_{\text{end}}}$ bezeichnet, wobei $X(t) = (x_{ij}(t))_{i,j=1,\dots,N}$ für $t = 0, \dots, t_{\text{end}}$. Unter der *Stichprobennahme* („sampling“) der *Kundenverteilung* wird die Monte-Carlo-Simulation von Szenarien der Kundenverteilung verstanden. Ein so erzeugtes Szenario wird auch *Stichprobe der Kundenverteilung* genannt. □

Anmerkungen

1. Ein stochastischer Prozess der Kundenverteilung ist ein stochastischer Prozess [8].
2. In der Literatur werden Szenarien auch als *Pfade* des stochastischen Prozesses bezeichnet.

6. Mit „Verteilung“ ist hier nicht die Verteilung aller Kunden im Akquiseplan auf die Knoten des Akquiseplans gemeint, sondern die Verteilung der den Knoten wechselnden Kunden auf die Folgeknoten.

4.3.3 Erstes und zweites Optimierungsproblem als ganzzahliges stochastisches lineares Optimierungsproblem

Die Optimierungsvariablen ergeben sich aus den verbalen Beschreibungen des ersten bzw. zweiten Optimierungsproblems (vgl. Abschnitt 3.4). Es sind die dem Akquiseplan zugeführten Kundenzahlen c_0 bzw. $c_0, c_1, \dots, c_{t_{\text{in}}}$. In Def. 4.18 wurde die Kundenverteilung, das heißt die Verteilung der den Knoten wechselnden Kunden auf die Folgeknoten so definiert, dass sie von den Optimierungsvariablen unabhängig ist. Hingegen wurde in Def. 4.12 die Anzahl der neuen Kunden in den Knoten des Akquiseplans als von den Optimierungsvariablen abhängige Zufallsvariable eingeführt. Deshalb sind diese Größen hier nicht brauchbar und werden in folgender Definition durch neue ersetzt.

Definition 4.19: *Anzahl der Kunden/Kundenanzahl zur Zeit t im Knoten A_i , Anzahl neuer Kunden zur Zeit t im Knoten A_i , Anzahl gegangener Kunden zur Zeit t im Knoten A_i , Anzahl der Kunden, die zur Zeit t vom Knoten A_i zum Knoten A_j gewechselt haben, Anzahl der Kunden, die zur Zeit t im Knoten A_i geblieben sind.*

Gegeben sei ein Akquiseplan AP mit den Knoten A_1, \dots, A_N , den Mengen $\text{Vorg}(j)$ und $\text{Nachf}(j)$ für $j = 1, \dots, N$ gemäß Def. 3.13, dem Akquisebeginn $t = 0$ und dem Akquiseende $t = t_{\text{end}}$. Dann bezeichnet

$$z(j, t) \in \mathbb{N}_0 \quad (4.71)$$

die Anzahl der Kunden/Kundenanzahl zur Zeit t im Knoten A_j ,

$$\Delta^+ z(j, t) \in \mathbb{N}_0 \quad (4.72)$$

die Anzahl neuer Kunden zur Zeit t im Knoten A_j , das ist die Anzahl der Kunden, die im Zeitraum $(t-1, t]$ von einem der Knoten A_i , $i \in \text{Vorg}(j)$ zum Knoten A_j wechseln, und

$$\Delta^- z(j, t) \in \mathbb{N}_0 \quad (4.73)$$

die Anzahl gegangener Kunden zur Zeit t im Knoten A_j , das ist die Anzahl der Kunden, die im Zeitraum $(t-1, t]$ vom Knoten A_j zu einem der Knoten A_k , $k \in \text{Nachf}(j)$ wechseln, jeweils für $t = 0, \dots, t_{\text{end}}$ und $j = 1, \dots, N$, und

$$\Delta z(i, j, t) \in \mathbb{N}_0 \quad (4.74)$$

die Anzahl der Kunden, die zur Zeit t vom Knoten A_i zum Knoten A_j gewechselt haben, das ist die Anzahl der Kunden, die im Zeitraum $(t-1, t]$ vom Knoten A_i zum Knoten A_j wechseln, für $t = 1, \dots, t_{\text{end}}$, $i = 1, \dots, N$ und $j \in \text{Nachf}(i)$, und

$$\Delta z(i, i, t) \in \mathbb{N}_0 \quad (4.75)$$

die Anzahl der Kunden, die zur Zeit $t - 1$ im Knoten A_i geblieben sind, das ist die Anzahl der Kunden, die im Zeitraum $(t - 1, t]$ im Knoten A_i bleiben, für $t = 1, \dots, t_{\text{end}}$ und $j = 1, \dots, N$. \square

Der folgende Hilfssatz wird weiter unten benötigt.

Hilfssatz 4.9: Betrachtet werde eine Akquise mit dem Akquiseplan AP mit der Knotenmenge $\underline{A} = \{A_1, \dots, A_N\}$, den Mengen $\text{Vorg}(j)$ und $\text{Nachf}(j)$ für $j = 1, \dots, N$ gemäß Def. 3.13 und dem Akquisezeitraum $T = \{0, 1, \dots, t_{\text{end}}\}$. Dann gilt

$$\sum_{i=1}^N \sum_{j \in \text{Nachf}(i)} \Delta z(i, j, t) = \sum_{j=1}^N \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t), \quad (4.76)$$

für alle $t \in T$. \square

Beweis: Der Beweis wird direkt geführt. Sei

$$\text{Nachf} := \bigcup_{i=1}^N \text{Nachf}(i).$$

Nach Vertauschen der Summen gilt damit für die linke Seite von (4.76):

$$\sum_{i=1}^N \sum_{j \in \text{Nachf}(i)} \Delta z(i, j, t) = \sum_{j \in \text{Nachf}} \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t). \quad (4.77)$$

Weil $\text{Nachf} \subseteq \{1, 2, \dots, N\}$ und $\text{Vorg}(j) = \emptyset$ für $j \in \{1, 2, \dots, N\} \setminus \text{Nachf}$ ist

$$\sum_{j \in \text{Nachf}} \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t) = \sum_{j=1}^N \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t),$$

woraus mit (4.77) die Behauptung (4.76) folgt. \blacksquare

4.3.3.1 Bilanzgleichungen der Kundenzahlen

Unter Berücksichtigung des stochastischen Prozesses der Kundenverteilung, $(\Xi(t))_{t=0}^{t_{\text{end}}}$ können für die Kundenanzahlen (4.71) – (4.75) Bilanzgleichungen aufgestellt werden:

$$z(j, t) = z(j, t - 1) + \Delta^+ z(j, t) - \Delta^- z(j, t) \quad (4.78)$$

für $j = 1, \dots, N$ und $t = 1, \dots, t_{\text{end}}$ und

$$z(1, 0) = c_0, \quad (4.79)$$

$$z(j, 0) = 0 \quad (4.80)$$

für $j = 2, \dots, N$ und

$$\Delta^+ z(j, t) = \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t) \quad (4.81)$$

für $j = 1, \dots, N$ und $t = 1, \dots, t_{\text{end}}$ und

$$\Delta^- z(j, t) = \sum_{k \in \text{Nachf}(j)} \Delta z(j, k, t) \quad (4.82)$$

für $j = 1, \dots, N$ und $t = 1, \dots, t_{\text{end}}$ und

$$\Delta z(j, j, t) = z(j, t-1) - \Delta^- z(j, t) \quad (4.83)$$

für $j = 1, \dots, N$ und $t = 1, \dots, t_{\text{end}}$ und

$$\xi_{ij}(t) \cdot z(i, t-1) - 1 \leq \Delta z(i, j, t) \leq \xi_{ij}(t) \cdot z(i, t-1) + 1 \quad (4.84)$$

für $j \in \overline{\text{Nachf}(i)}$.

Anmerkungen

1. Gleichung 4.78 ist die grundlegende Bilanzgleichung. Sie sagt aus, dass sich die Anzahl der Kunden, die sich zum Zeitpunkt t im Knoten A_j befindet, aus drei Bestandteilen zusammensetzt: die Kunden, die bereits zum Zeitpunkt $t-1$ im Knoten A_j waren und die Kunden, die im Zeitraum $(t-1, t]$ aus einem Vorgängerknoten von A_j zu dem Knoten A_j gewechselt sind, und davon ist – als negativer „Bestandteil“ – die Anzahl der Kunden abzuziehen, die im Zeitraum $(t-1, t]$ vom Knoten A_j zu einem seiner Nachfolgerknoten wechselt. Weil (4.78) nur ab $t = 1$ gilt, sind die Anfangswerte $z(j, 0)$ für alle Knoten A_j zu initialisieren. Dies geschieht in den Gleichungen (4.79) und (4.80).
2. Der Zuwachs bzw. Weggang an Kunden im Zeitraum $(t-1, t]$ in bzw. von einem Knoten setzt sich aus den Kunden zusammen, die in diesem Zeitraum von einem Knoten A_i zu einem anderen Knoten A_j wechseln. Das sagen die Gleichungen (4.81) und (4.82) aus.
3. Gleichung (4.83) besagt, dass die Anzahl der Kunden, die in einem Knoten bleiben gleich der Anzahl der Kunden in diesem Knoten abzüglich der Anzahl der Kunden, die den Knoten verlassen, ist.
4. Die von einem Knoten A_i zu einem anderen Knoten A_j wechselnden bzw. in ihm bleibenden Kundenzahlen werden von den Zufallsvariablen $\xi_{ij}(t)$ bestimmt. Die Zufallsvektoren

$(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}}$ gehorchen einer normierten Multinomialverteilung $M_{c, (\Psi_{ij}^t)_{j \in \overline{\text{Nachf}(i)}}}^n$ und nehmen deshalb, durch den Parameter c festgelegt, nur bestimmte rationale Werte im Intervall $[0, 1]$ an. Daher wird das Produkt $\xi_{ij}(t) \cdot z(i, t-1)$ in (4.84) in der Regel nicht ganzzahlig, sondern gebrochen sein. Die Ungleichungen in (4.84) gewährleisten, dass $\Delta z(i, j, t)$ einen ganzzahligen Wert annehmen kann, der entweder dem ganzzahligen Anteil des Wertes $\lfloor \xi_{ij}(t) \cdot z(i, t-1) \rfloor$ entspricht, oder um eins höher oder um eins kleiner als dieser Wert sein kann. Diese Variationsbreite ist notwendig. Es reicht nicht aus

$$\text{rd}[\xi_{ij}(t) \cdot z(i, t-1)] = \Delta z(i, j, t)$$

zu fordern, unabhängig von der Definition des Rundungsoperators. Dies verdeutlicht folgendes Beispiel:

Ohne Beschränkung der Allgemeinheit sei $i = 1$, $\overline{\text{Nachf}(1)} = \{1, 2, 3\}$, $z(1, t-1) = 2$ und $\xi_{1j} = 1/3$ für $j = 1, 2, 3$. Würde der Rundungsoperator abrunden, dann wäre

$$\text{rd}[\xi_{1j}(t) \cdot z(1, t-1)] = \text{rd}[2/3] = 0 = \Delta z(1, j, t)$$

und $\sum_{j=1}^3 \Delta z(1, j, t) = 0$. Würde der Rundungsoperator aufrunden oder kaufmännisch runden, dann wäre

$$\text{rd}[\xi_{1j}(t) \cdot z(1, t-1)] = \text{rd}[2/3] = 1 = \Delta z(1, j, t)$$

und $\sum_{j=1}^3 \Delta z(1, j, t) = 3$. In beiden Fällen hat $\sum_{j=1}^3 \Delta z(1, j, t)$ nicht den gewünschten Wert zwei, der aus (4.82) und (4.83) folgt. Mit (4.84) können die $\Delta z(1, j, t)$ Werte aus dem Intervall $[-1/3, 5/3]$ annehmen, also auch die zulässigen ganzzahligen Kombinationen $(0, 1, 1)$, $(1, 0, 1)$ oder $(1, 1, 0)$. Die genaue Festlegung bleibt dem Optimierungsalgorithmus überlassen.

5. Die Gesamtzahl der im Akquiseplan vorhandenen Kunden ist für alle $t = 0, 1, \dots, t_{\text{end}}$ konstant, denn aus (4.78), (4.81) und (4.82) folgt mit Hilfssatz 4.9:

$$\begin{aligned} \sum_{j=1}^N z(j, t) &= \sum_{j=1}^N z(j, t-1) + \sum_{j=1}^N \Delta^+ z(j, t) - \sum_{j=1}^N \Delta^- z(j, t) \\ &= \sum_{j=1}^N z(j, t-1) + \underbrace{\sum_{j=1}^N \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t) - \sum_{j=1}^N \sum_{k \in \text{Nachf}(j)} \Delta z(j, k, t)}_{= 0} = \sum_{j=1}^N z(j, t-1). \end{aligned}$$

4.3.3.2 Erstes Optimierungsproblem

Das erste Optimierungsproblem lautet (vgl. Abschnitt 3.4):

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan gleichzeitig unter Einhaltung der Kapazitätsschranken der Akquiseaktionen zugeführt werden kann?

Daraus folgt die Zielfunktion

$$c_0 \stackrel{!}{=} \max.$$

In die Formulierung des ersten Optimierungsproblems gehen die Gleichungen (4.78) – (4.84) ein. Neben den Bilanzgleichungen wird die Menge der zulässigen Punkte des Optimierungsproblems durch die Kapazitätsbeschränkungen in den Knoten bestimmt. Für die unscharfen Kapazitätsschranken l_j und das Wahrheitsniveau δ_l lauten sie

$$\Delta^+ z(j, t) \leq \mu_{l_j}^{-1}(\delta_l)$$

für $j = 1, \dots, N$, $t = 0, 1, \dots, t_{\text{end}}$, wobei $\mu_{l_j}^{-1}$ die verallgemeinerte Umkehrabbildung der unscharfen Kapazitätsschranke l_j nach Def. 4.3 ist. Jetzt kann das erste Optimierungsproblem definiert werden.

Definition 4.20: *Erstes Optimierungsproblem.*

Gegeben sei ein Akquiseplan AP mit den Knoten A_1, \dots, A_N , den Mengen $\text{Vorg}(i)$ und $\text{Nachf}(i)$ bzw. $\overline{\text{Nachf}}(i)$ für $i = 1, \dots, N$ gemäß Def. 3.13, einer Akquise, die zum Zeitpunkt $t = 0$ beginnt und in $t = t_{\text{end}}$ endet und dem stochastischen Prozess der Kundenverteilung $(\Xi(t))_{t=0}^{t_{\text{end}}}$. Dann lautet das *erste Optimierungsproblem* zum Wahrheitsniveau $\delta_{l_j} \in (0, 1]$:

$$c_0 \stackrel{!}{=} \max$$

unter den Nebenbedingungen (für die Größen aus Def. 4.19):

$$z(j, t) = z(j, t-1) + \Delta^+ z(j, t) - \Delta^- z(j, t) \text{ für } j = 1, \dots, N,$$

$$z(1, 0) = c_0 \in \mathbb{N}_0, z(j, 0) = 0 \text{ für } j = 2, \dots, N,$$

$$\Delta^+ z(j, t) = \sum_{i \in \text{Vorg}(j)} \Delta z(i, j, t) \text{ für } j = 1, \dots, N,$$

$$\Delta^- z(j, t) = \sum_{k \in \text{Nachf}(j)} \Delta z(j, k, t) \text{ für } j = 1, \dots, N,$$

$$z(j, j, t) = z(j, t-1) - \Delta^- z(j, t) \text{ für } j = 1, \dots, N,$$

$$\Delta^+ z(j, t) \leq \mu_{l_j}^{-1}(\delta_l) \text{ für } j = 2, \dots, N,$$

$$\xi_{ij}(t) \cdot z(i, t-1) - 1 \leq \Delta z(i, j, t) \leq \xi_{ij}(t) \cdot z(i, t-1) + 1 \text{ mit } j \in \overline{\text{Nachf}}(i) \\ \text{für } i = 1, \dots, N,$$

$$z(j, t), \Delta z(j, k, t), \Delta^+ z(j, t), \Delta^- z(j, t) \in \mathbb{N}_0 \text{ für } j = 1, \dots, N, \\ k \in \overline{\text{Nachf}(j)},$$

jeweils für $t = 1, \dots, t_{\text{end}}$.

□

Anmerkungen

1. Das erste Optimierungsproblem kann in der in Def. 4.20 angegebenen Form mit moderner Software, z. B. der *C++-Schnittstelle* für *CPLEX*⁷⁾ direkt in ein Computerprogramm umgesetzt werden. Die Überführung in eine Standardform ist nicht erforderlich.
2. Die Optimierungsvariablen des Problems sind:

$$z(j, t), \Delta z(j, k, t), \Delta^+ z(j, t), \Delta^- z(j, t) \in \mathbb{N}_0$$

für $j = 1, \dots, N$, $k \in \overline{\text{Nachf}(j)}$ und für $t = 1, \dots, t_{\text{end}}$ sowie

$$z(j, 0)$$

für $j = 1, \dots, N$, wobei nur $z(1, 0) = c_0$ einen Zielfunktionskoeffizienten ungleich null besitzt.

3. Das erste Optimierungsproblem ist ein *ganzzahliges stochastisches lineares Optimierungsproblem*.

4.3.3.3 Zweites Optimierungsproblem

Das zweite Optimierungsproblem lautet (vgl. Abschnitt 3.4):

Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan innerhalb eines gegebenen Zeitraums unter Einhaltung der Kapazitätsschranken für Akquiseaktionen zugeführt werden kann, und wie ist sie über den Zeitraum verteilt?

Das zweite Optimierungsproblem geht davon aus, dass $c_\tau \in \mathbb{N}_0$ Kunden dem Akquiseplan für jedes $\tau = 0, 1, \dots, t_{\text{in}}$ zugeführt werden. Daraus folgt die Zielfunktion

$$\sum_{\tau=0}^{t_{\text{in}}} c_\tau \stackrel{!}{=} \max.$$

Unter der Annahme, dass sich Kunden unabhängig voneinander durch den Akquiseplan bewegen, kann die Verteilung der Kunden auf die Knoten im Akquiseplan für jeden Zuführungszeitpunkt getrennt betrachtet werden („Superposition“). Die Größen aus Def. 4.19 werden durch einen Index entsprechend gekennzeichnet, das heißt

$$z_\tau(j, t) \in \mathbb{N}_0 \text{ entsprechen (4.71),}$$

7. siehe Fußnote 26 auf Seite 168

$$\Delta^+ z_\tau(j, t) \in \mathbb{N}_0 \text{ entsprechen (4.72),}$$

$$\Delta^- z_\tau(j, t) \in \mathbb{N}_0 \text{ entsprechen (4.73),}$$

$$\Delta z_\tau(j, k, t) \in \mathbb{N}_0 \text{ entsprechen (4.74),}$$

jeweils für $\tau = 0, 1, \dots, t_{\text{in}}$, $t = \tau, \tau + 1, \dots, t_{\text{end}}$, $j = 1, \dots, N$, $k \in \text{Nachf}(j)$. Damit lautet das zweite Optimierungsproblem:

Definition 4.21: *Zweites Optimierungsproblem.*

Gegeben sei ein Akquiseplan AP mit den Knoten A_1, \dots, A_N , den Mengen $\text{Vorg}(i)$ und $\text{Nachf}(i)$ bzw. $\overline{\text{Nachf}}(i)$ für $i = 1, \dots, N$ gemäß Def. 3.13, einer Akquise, die zum Zeitpunkt $t = 0$ beginnt und in $t = t_{\text{end}}$ endet und dem stochastischen Prozess der Kundenverteilung $(\Xi(t))_{t=0}^{t_{\text{end}}}$. Wenn Kunden zu den Zeitpunkten $\tau = 0, 1, \dots, t_{\text{in}}$, $t_{\text{in}} \ll t_{\text{end}}$ dem Akquiseplan zugeführt werden können, dann lautet das *zweite Optimierungsproblem* zum Wahrheitsniveau $\delta_1 \in (0, 1]$:

$$\sum_{\tau=0}^{t_{\text{in}}} c_\tau \stackrel{!}{=} \max$$

unter den Nebenbedingungen:

$$z_\tau(j, t) = z_\tau(j, t-1) + \Delta^+ z_\tau(j, t) - \Delta^- z_\tau(j, t) \text{ für } j = 1, \dots, N, \quad (4.85)$$

$$z_\tau(1, \tau) = c_\tau \in \mathbb{N}_0, z_\tau(j, \tau) = 0 \text{ für } j = 2, \dots, N, \quad (4.86)$$

$$\Delta^+ z_\tau(j, t) = \sum_{i \in \text{Vorg}(j)} \Delta z_\tau(i, j, t) \text{ für } j = 1, \dots, N, \quad (4.87)$$

$$\Delta^- z_\tau(j, t) = \sum_{k \in \text{Nachf}(j)} \Delta z_\tau(j, k, t) \text{ für } j = 1, \dots, N, \quad (4.88)$$

$$\Delta z_\tau(j, j, t) = z_\tau(j, t-1) - \Delta^- z_\tau(j, t) \text{ für } j = 1, \dots, N, \quad (4.89)$$

$$\begin{aligned} \xi_{ij}(t) \cdot z(i, t-1) - 1 &\leq \Delta z(i, j, t) \leq \xi_{ij}(t) \cdot z(i, t-1) + 1 \\ \text{mit } j &\in \overline{\text{Nachf}}(i) \text{ für } i = 1, \dots, N, \end{aligned} \quad (4.90)$$

$$z_\tau(j, t), \Delta z_\tau(j, k, t), \Delta^+ z_\tau(j, t), \Delta^- z_\tau(j, t) \in \mathbb{N}_0 \text{ mit } k \in \overline{\text{Nachf}}(j) \text{ für } j = 1, \dots, N,$$

jeweils für $\tau = 0, 1, \dots, t_{\text{in}}$ und $t = \tau + 1, \dots, t_{\text{end}}$ und

$$\sum_{\tau=0}^{\min\{t_{\text{in}}, t\}} \Delta^+ z_{\tau}(j, t) \leq \mu_j^{-1}(\delta_j) \text{ für } j = 1, \dots, N, \quad (4.91)$$

für $t = 0, 1, \dots, t_{\text{end}}$.

□

Gegenüber dem ersten Optimierungsproblem ändern sich die Zielfunktion und die Kapazitätsnebenbedingungen. Die Struktur eines *ganzzahligen stochastischen linearen Optimierungsproblems* bleibt erhalten.

4.3.4 Zweites Optimierungsproblem als Mehrgüterfluss-Problem

Bereits im Abschnitt 4.2.3.2 wurde darauf hingewiesen, dass ein allgemeiner Lösungsansatz für ganzzahlige Optimierungsprobleme darin besteht, ein reellwertiges Optimierungsproblem zu bestimmen, das eine optimale Lösung besitzt, die zugleich eine optimale Lösung des ganzzahligen Problems ist. Ist das reellwertige „Ersatzproblem“ bekannt, dann wird dieses mit einem der bekannten Verfahren gelöst, z. B., wenn das Optimierungsproblem linear ist, mit dem Simplexverfahren [117]. Für das zweite Optimierungsproblem aus dem vorigen Abschnitt wird dieses Verfahren gewählt, weil die große Anzahl der Optimierungsvariablen (vgl. Def. 4.21) bei der Lösung mit einem Verfahren zur ganzzahligen Optimierung zu Laufzeitproblemen führen könnte. Auch bei einem überschaubaren Akquiseplan mit etwa 20 Knoten wird das zweite Optimierungsproblem für einen Initialisierungszeitraum mit zehn Zeitschritten und einen Akquisezeitraum von 100 Zeitschritten mehrere zehntausend Variablen besitzen.

Nach Einführung einiger grundlegender Begriffe im Abschnitt 4.3.4.1 wird das zweite Optimierungsproblem im Abschnitt 4.3.4.2 als *maximaler-Fluss-Problem* beschrieben, weil maximaler-Fluss-Probleme bei ganzzahligen Kantenkapazitäten eine ganzzahlige Lösung besitzen [120]. Jedoch können einige der Nebenbedingungen aus der Formulierung des zweiten Optimierungsproblems gemäß Def. 4.21 nicht in die Formulierung des maximaler-Fluss-Problems übertragen werden. Die Lösung zur Handhabung dieser „unbequemen Nebenbedingungen“ besteht in der Lagrange-Relaxation des Problems [120], die Gegenstand von Abschnitt 4.3.4.3 ist.

4.3.4.1 Grundlagen

Zunächst sind in Anlehnung an [17] einige Begriffe zu definieren:

Definition 4.22: *Flussvektor eines Graphen, Fluss einer Kante, Divergenz eines Graphen, Divergenz eines Knotens, Quellenstärke, Senkenstärke, Quelle, Senke, Fluss des Graphen.*

Der *Flussvektor des Graphen* $G := (\underline{A}, \underline{R})$ mit der Knotenmenge $\underline{A} = \{A_1, \dots, A_N\}$ und der Kantenmenge \underline{R} ist eine Menge von Skalaren,

$$\{x_{ij} \in \mathbb{R} \mid (A_i, A_j) \in \underline{R}\},$$

wobei x_{ij} als *Fluss der Kante* (A_i, A_j) bezeichnet wird. Die *Divergenz des Graphen* bezüglich des Flussvektors ist der N -dimensionale Vektor $y = (y_1, \dots, y_N)$ mit den Elementen

$$y_j := \sum_{k \in \text{Nachf}(j)} x_{jk} - \sum_{i \in \text{Vorg}(j)} x_{ij}$$

für $j = 1, \dots, N$. Ein Element y_j wird *Divergenz (des Graphen) im Knoten* A_j genannt. Eine positive Divergenz heißt *Quellenstärke*, eine negative Divergenz *Senkenstärke*⁸⁾. Ein Knoten, von dem nur Kanten ausgehen, wird *Quelle*, ein Knoten, in dem nur Kanten enden, wird *Senke* genannt⁹⁾. Besitzt der Graph G genau eine Quelle und eine Senke,

$$A_q, A_s \in \underline{A},$$

und wird der Graph (gedanklich) um die Kante (A_s, A_q) erweitert, dann wird der (gedachte) Fluss x_{sq} dieser Kante als *Fluss des Graphen* bezeichnet. \square

Anmerkungen

1. Die Summe der Knotendivergenzen eines Graphen ist stets null.
2. Insbesondere ist für einen Graphen mit genau einer Quelle und einer Senke

$$x_{sq} := \sum_{i \in \text{Vorg}(s)} x_{is} = \sum_{k \in \text{Nachf}(q)} x_{qk}$$

[17].

Das *maximaler-Fluss-Problem* wird in der Gestalt eines *minimale-Kosten-Fluss-Problems* formuliert. Werden die Kanten eines Graphen G (mit Gewichten) bewertet, die die pro-Einheit-Kosten des Kantenflusses darstellen, dann besteht das *minimale-Kosten-Fluss-Problem* für den Graphen darin, für einen gegebenen Fluss, die Summe der Kosten zu minimieren.

Definition 4.23: *Minimaler-Kosten-Fluss-Problem.*

Für den Graphen $G := (\underline{A}, \underline{R})$ mit dem Flussvektor

$$\{x_{ij} \in \mathbb{R} \mid (A_i, A_j) \in \underline{R}\},$$

den Knoten-Divergenzen

$$q_i \text{ für } i = 1, \dots, N,$$

den Fluss-Beschränkungen der Kanten

-
8. Diese Begriffe sind der elektromagnetischen Feldtheorie entliehen [148].
 9. Diese Definition weicht von der in der Netzwerktheorie üblichen Definition ab, die Quellen und Senken allein über das Vorzeichen der Divergenz des betrachteten Knotens definieren (vgl. [17]). Sie ist hier jedoch zweckmäßig.

$$a_{ij} \leq x_{ij} \leq b_{ij} \text{ für } (A_i, A_j) \in \underline{R}$$

und den Kantenbewertungen

$$c_{ij} \text{ für } (A_i, A_j) \in \underline{R}$$

lautet das *Minimaler-Kosten-Fluss-Problem*:

$$\sum_{(A_i, A_j) \in \underline{R}} c_{ij} x_{ij} \stackrel{!}{=} \min$$

unter Einhaltung der Randbedingungen

$$\sum_{k \in \text{Nachf}(j)} x_{jk} - \sum_{i \in \text{Vorg}(j)} x_{ij} = q_j, A_j \in \underline{A} \quad (4.92)$$

und

$$a_{ij} \leq x_{ij} \leq b_{ij} \text{ für } (A_i, A_j) \in \underline{R}. \quad (4.93)$$

□

Das Maximaler-Fluss-Problem für den Graph G besteht in der Maximierung des Flusses des Graphen unter Einhaltung vorgegebener Beschränkungen für die Flüsse seiner Kanten.

Definition 4.24: *Maximaler-Fluss-Problem.*

Für den Graphen $G := (\underline{A}, \underline{R})$ mit der Quelle A_q und der Senke A_s , dem Flussvektor

$$\{x_{ij} \in \mathbb{R} \mid (A_i, A_j) \in \underline{R}\}$$

und den Fluss-Beschränkungen der Kanten

$$a_{ij} \leq x_{ij} \leq b_{ij} \text{ für alle } (A_i, A_j) \in \underline{R}$$

lautet das *Maximaler-Fluss-Problem* in der Formulierung eines *Minimale-Kosten-Fluss-Problems*:

$$-x_{sq} \stackrel{!}{=} \min \quad (4.94)$$

unter Einhaltung der Randbedingungen

$$\sum_{k \in \text{Nachf}(j)} x_{jk} - \sum_{i \in \text{Vorg}(j)} x_{ij} = 0, A_j \in \underline{A} \quad (4.95)$$

und

$$x_{sq} := \sum_{i \in \text{Vorg}(s)} x_{is} = \sum_{k \in \text{Nachf}(q)} x_{qk}$$

und

$$a_{ij} \leq x_{ij} \leq b_{ij}, (A_i, A_j) \in \underline{R} \quad (4.96)$$

und

$$\sum_{i \in \text{Vorg}(q)} a_{iq} \leq x_{sq} \leq \sum_{i \in \text{Vorg}(q)} b_{iq}. \quad (4.97)$$

□

Anmerkungen

1. (4.94) ist äquivalent zur Maximierung des Flusses im Graphen, $x_{sq} : x_{sq} \stackrel{!}{=} \max$.
2. Die Kantenbewertungen sind $c_{ij} = 0$ für $(A_i, A_j) \in \underline{R}$ und $c_{sq} = -1$.
3. Die Beschränkung (4.97) für die (gedachte) Kante (A_s, A_q) ist redundant und hier nur aufgeführt, um der formalen Beschreibung eines Minimale-Kosten-Fluss-Problems zu genügen.
4. Nebenbedingungen der Gestalt (4.92) und (4.95) sind Bilanzgleichungen der Flüsse in den Knoten und werden als *Fluss-Erhaltungs-Nebenbedingungen* bezeichnet.

Für die Beschreibung des zweiten Optimierungsproblems als Maximaler-Fluss-Problem ist der Graph des Akquiseplans nicht ausreichend. Die Ursache liegt darin, dass ein Akquiseplan die zeitliche Abfolge der Vorgänge bei einer Akquise nur qualitativ beschreibt, nicht aber quantitativ. Zur Behebung dieses Mangels wird in Def. 4.26 ein erweiterter Akquiseplan eingeführt, der *zeitlicher Akquiseplan* genannt wird. Zu seiner Definition ist der Begriff der τ -Menge nötig.

Definition 4.25: τ -Menge zur Zeit t und zum Knoten A_i .

Gegeben sei ein Akquiseplan AP mit dem Akquisebeginn $t = 0$ und dem Akquiseende $t = t_{\text{end}}$. Dann wird die Menge

$$T_{(i,t)} := \{\tau \in \{0, 1, \dots, t_{\text{in}}\} : d_q(i) \leq t - \tau \text{ und } d_s(i) \leq t_{\text{end}} - \max\{t, \tau\}\} \quad (4.98)$$

als τ -Menge zur Zeit t und zum Knoten A_i bezeichnet.

□

4.3.4.2 Maximaler-Fluss-Problem mit unbequemen Nebenbedingungen

Definition 4.26: Zeitlicher Akquiseplan.

Durchgeführt werde eine Akquise mit einem Akquiseplan $AP = (\underline{A}, \underline{R})$ (mit der Knotenmenge $\underline{A} = \{A_1, \dots, A_N\}$, der Quelle $A_q =: A_1$ und den Senken $A_p =: A_N$ und $A_n =: A_{N-1}$ gemäß Def. 3.1) und dem Akquisezeitraum $T = \{0, 1, \dots, t_{\text{in}}, \dots, t_{\text{end}}\}$ mit Kundenzuführung für $\tau = 0, 1, \dots, t_{\text{in}}$ für

$t_{\text{in}} \ll t_{\text{end}}$. Dann ist der zugehörige *zeitliche Akquiseplan* ein Graph $AP^T = (\underline{A}^T, \underline{R}^T)$ mit folgenden Eigenschaften:

Die *Knotenmenge* \underline{A}^T enthält folgende Knoten:
genau eine Quelle und genau eine Senke,

$$A_0 =: A_{(0, -1, -1)}, A_{N+1} =: A_{(N+1, -1, -1)} \quad (4.99)$$

sowie die Knoten

$$A_{(i, t, \tau)} \quad (4.100)$$

für alle $A_i \in \underline{A}$, $t \in T$ und $\tau \in T_{(i, t)}$.

Die *Kantenmenge* \underline{R}^T enthält folgende Kanten:

$$(A_0, A_{(1, \tau, \tau)}) \quad (4.101)$$

für $\tau = 0, 1, \dots, t_{\text{in}}$,

$$(A_{(N, t, \tau)}, A_{N+1}) \text{ und } (A_{(N-1, t, \tau)}, A_{N+1}) \quad (4.102)$$

für $A_{(N, t, \tau)}, A_{(N-1, t, \tau)} \in \underline{A}^T$,

$$(A_{(i, t-1, \tau)}, A_{(j, t, \tau)}) \quad (4.103)$$

für $A_{(i, t-1, \tau)}, A_{(j, t, \tau)} \in \underline{A}^T$ und $(A_i, A_j) \in \underline{R}$,

$$(A_{(i, t-1, \tau)}, A_{(i, t, \tau)}) \quad (4.104)$$

für $A_{(i, t-1, \tau)}, A_{(i, t, \tau)} \in \underline{A}^T$. □

Anmerkungen

1. Da ein Akquiseplan gemäß Def. 3.1 zwei Senken, A_p und A_n besitzt, wird für den zeitlichen Akquiseplan eine neue Senke, A_{N+1} , (4.99) eingeführt, die A_p und A_n zusammenfasst.
2. Der einheitlichen Notation halber werden für A_0 und A_{N+1} auch die Bezeichnungen

$$A_{(0, -1, -1)} := A_0$$

und

$$A_{(N+1, -1, -1)} := A_{N+1}$$

verwendet (Glg. 4.100).

3. Ein zeitlicher Akquiseplan besteht aus $t_{\text{in}} + 1$ induzierten Untergraphen¹⁰⁾, jeweils einer für $\tau = 0, 1, \dots, t_{\text{in}}$, die nur die Quelle A_0 und die Senke A_{N+1} gemeinsam haben.
4. Man kann sich die Knoten (4.100) eines zeitlichen Akquiseplans schichtenweise angeordnet denken, wobei jede Schicht einem Zeitpunkt $t \in T$ entspricht. Die Menge der Knoten mit gleichem Indexbestandteil t wird im Folgenden auch als *Zeitschicht* bezeichnet. Diese Schichtung ist auch bei jedem der induzierten Untergraphen aus 3. vorhanden. Die Quelle und die Senke, A_0 und A_{N+1} sind keiner Zeitschicht zugeordnet.
5. Zur Vereinfachung der Schreibweise werden Knoten $A_{(i, t, \tau)}$ im Folgenden mit ihrem Indextripel (i, t, τ) gleichgesetzt und zur Bezeichnung wird nur das Indextripel verwendet.
6. Bei der Indizierung der Knoten (4.100) wurde beachtet, dass sich Kunden in einem Zeitschritt, $t \rightarrow t + 1$ höchstens um einen Knoten weiter bewegen können. Deshalb sind im Index (i, t, τ) nicht alle Wertekombinationen der Bestandteile i , t und τ möglich. Ihr Zusammenhang wird durch (4.98) in Def. 4.25 ausgedrückt.
7. Aufgrund von (4.98) nimmt die Anzahl der Zeitschichten der induzierten Untergraphen aus 3. mit τ ab. Der induzierte Untergraph für $\tau = 0$ besitzt $t_{\text{end}} + 1$ Zeitschichten, der induzierte Untergraph für $\tau = t_{\text{in}}$ besitzt $t_{\text{end}} - t_{\text{in}} + 1$ Zeitschichten.
8. Damit die dem Akquiseplan AP zugeführten Kundenzahlen $c_0, c_1, \dots, c_{t_{\text{in}}}$ jeweils als Kantentröffe dargestellt werden können, wird der neue Quellknoten A_0 in (4.99) eingeführt. Über die von ihm ausgehenden Kanten (4.101) führen diese Kundenflüsse zu den Zeitschichten $0, 1, \dots, t_{\text{in}}$.
9. Entsprechend (4.98) in Def. 4.25 enthält die Zeitschicht $t = 0$ nur den Knoten $A_{(1, 0, 0)}$.
10. Die Kanten (4.102) verbinden die Senken $A_{(N, t, \tau)}$ und $A_{(N-1, t, \tau)}$ der Zeitschichten t mit der „globalen“ Senke A_{N+1} .
11. Kundenbewegungen von einem Knoten zu einem seiner Folgeknoten werden im zeitlichen Akquiseplan als Flüsse dargestellt. In (4.103) stehen die Kanten, über die sie führen. Die zu unterschiedlichen Zeitpunkten zugeführten Kunden bilden unabhängige Flüsse durch den zeitlichen Akquiseplan. Mit Ausnahme der Kanten von der Quelle bzw. zur Senke des zeitlichen Akquiseplans werden deshalb nur Knoten mit gleichem Index τ durch Kanten verbunden.
12. Aufgrund der Aufteilung eines Knotens A_i des Akquiseplans AP auf mehrere Zeitschichten im zeitlichen Akquiseplan AP^T können die Kunden, die in einem Zeitschritt $t \rightarrow t + 1$ im Knoten (des Akquiseplans AP) bleiben, im Akquiseplan AP^T als Fluss auf den Kanten

10. Ein Graph $H = (E', K')$ heißt ein Untergraph von $G = (E, K)$, falls $E' \subseteq E$ und $K' \subseteq K$. H heißt ein induzierter Untergraph, falls H alle Kanten zwischen den Ecken in E' enthält, die auch in G vorhanden sind [1].

(4.104) dargestellt werden. Dabei wird mit dem Indexbestandteil τ zwischen Kunden unterschieden, die dem Akquiseplan zu unterschiedlichen Zeitpunkten zugeführt werden.

13. Zur Vereinfachung der Schreibweise werden die Kanten (4.103) und (4.104), $(A_{(i, t-1, \tau)}, A_{(j, t, \tau)})$ im Folgenden mit ihrem Indextripel-Paar $((i, t-1, \tau), (j, t, \tau))$ gleichgesetzt und dieses der Kürze halber mit dem Quadrupel (i, j, t, τ) bezeichnet. Kanten, die vom Quellknoten A_0 zu den Knoten $A_{(1, \tau, \tau)} \in \underline{A}$ ausgehen, (4.101), erhalten die Bezeichnung

$$(0, 1, \tau, \tau). \quad (4.105)$$

Kanten, die die Senken der einzelnen Zeitschichten, $A_{(N-1, t, \tau)}, A_{(N, t, \tau)} \in \underline{A}$ mit der „globalen Senke“ A_{N+1} verbinden, (4.102), erhalten die Bezeichnung

$$(N-1, N+1, t, \tau) \text{ bzw. } (N, N+1, t, \tau). \quad (4.106)$$

14. Zur Veranschaulichung des Zusammenhangs zwischen AP , T , t_{in} und AP^T zeigt Abb. 4.4 einen Akquiseplan und Abb. 4.5 den zugehörigen zeitlichen Akquiseplan.

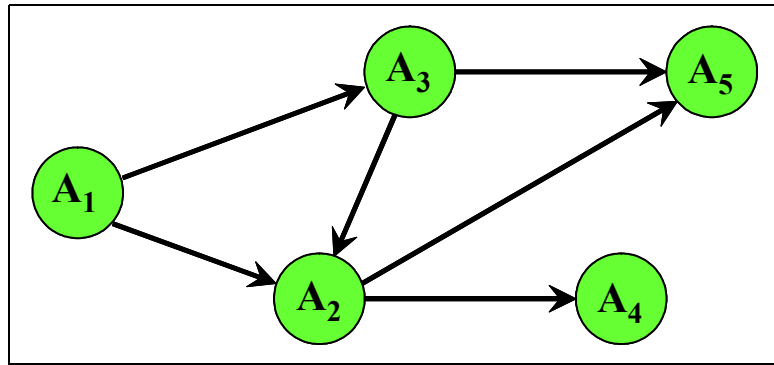


Abbildung 4.4: Akquiseplan AP mit $A_q = A_1$, $A_n = A_4$ und $A_p = A_5$.

Vor weiteren Betrachtungen zum Akquiseplan werden, zur vereinfachten Formulierung einiger Zusammenhänge, in Def. 4.27 Mengen von Vorgänger- und Nachfolgerknoten des zeitlichen Akquiseplans eingeführt.

Definition 4.27: (vollständige) Menge der Vorgängerknoten, (vollständige) Menge der Nachfolgerknoten im zeitlichen Akquiseplan.

Gegeben sei ein zeitlicher Akquiseplan $AP^T = (\underline{A}^T, \underline{R}^T)$ gemäß Def. 4.26. Zu jedem Knoten $A_{(i, t, \tau)} \in \underline{A}^T$ gibt es eine Menge von Vorgängerknoten ,

$$\text{Vorg}_T(N+1, -1, -1) := \{A_{(N, t, \tau)}, A_{(N-1, t, \tau)} \in \underline{A}^T\},$$

$$\text{Vorg}_T(j, t, \tau) := \{A_{(i, t-1, \tau)} \in \underline{A}^T : (i, j, t, \tau) \in \underline{R}^T \wedge i \neq j\} \text{ für } (j, t, \tau) \neq (N+1, -1, -1)$$

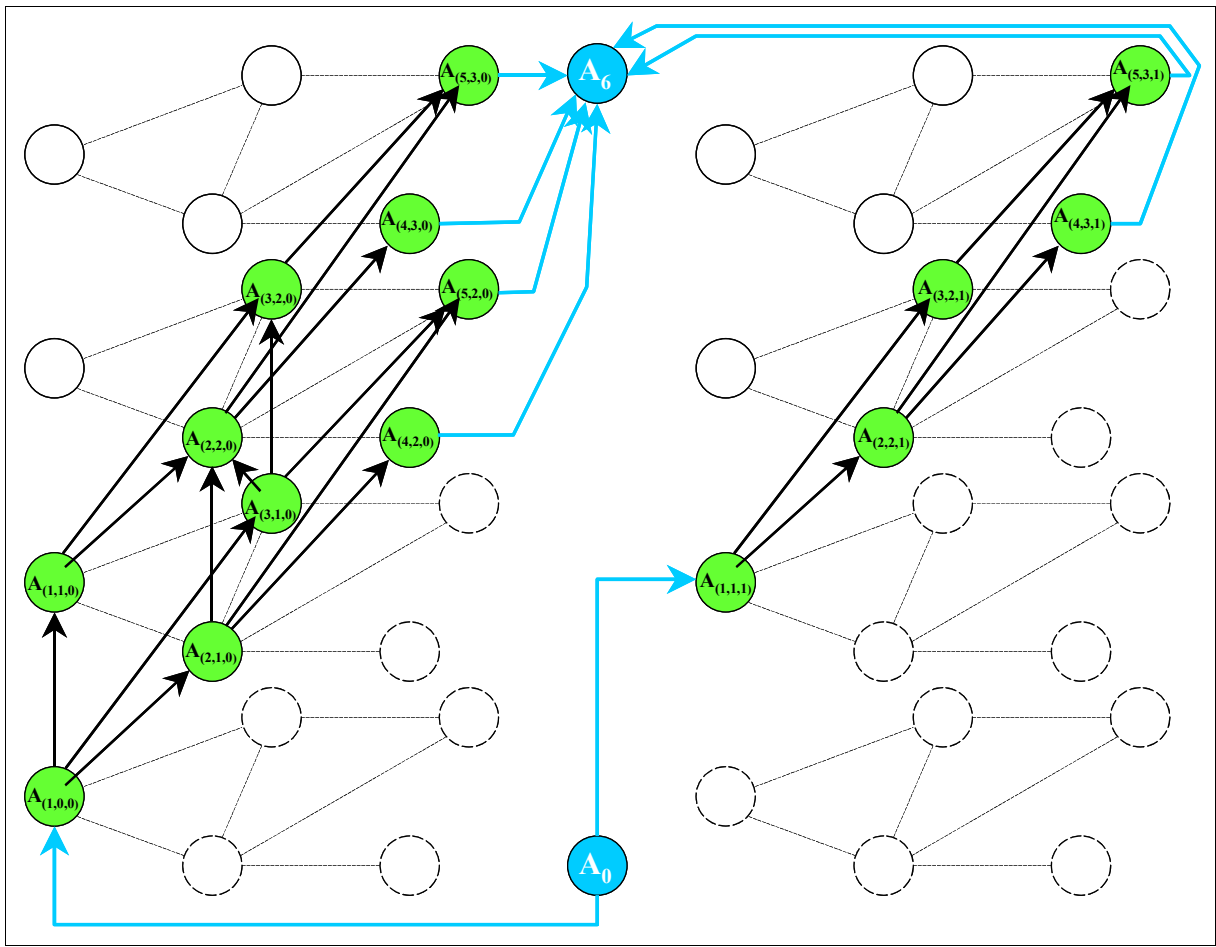


Abbildung 4.5: Zeitlicher Akquiseplan AP^T zum Akquiseplan AP in Abb. 4.4 mit Kundenzuführung für $\tau = 0, 1$ ($t_{\text{in}} = 1$) und Akquisezeitraum $T = \{0, 1, \dots, 5\}$ ($t_{\text{end}} = 5$).

und eine Menge von Nachfolgerknoten,

$$\text{Nachf}_T(0, -1, -1) := \{A_{(1, \tau, \tau)} : \tau = 0, \dots, t_{\text{in}}\},$$

$$\text{Nachf}_T(j, t, \tau) := \{A_{(k, t+1, \tau)} \in \underline{A}^T : (j, k, t+1, \tau) \in \underline{R}^T \wedge i \neq j\} \text{ für } (j, t, \tau) \neq (0, -1, -1)$$

sowie eine vollständige Menge von Vorgänger-knoten,

$$\overline{\text{Vorg}}_T(j, t, \tau) := \text{Vorg}_T(j, t, \tau) \cup \{A_{(j, t, \tau)}\}$$

und eine vollständige Menge von Nachfolgerknoten,

$$\overline{\text{Nachf}}_T(j, t, \tau) := \text{Nachf}_T(j, t, \tau) \cup \{A_{(j, t, \tau)}\}.$$

Diese Mengen können auch leer sein. Der „Index“ 'T' dient nur der Unterscheidung von den gleichnamigen Mengen aus Def. 3.13. \square

Mit der Definition des zeitlichen Akquiseplans steht nun ein Graph zur Verfügung, der die Formulierung des zweiten Optimierungsproblems (Def. 4.21) als Maximaler-Fluss-Problem in der Formulierung eines Minimale-Kosten-Fluss-Problems gestattet (vgl. Def. 4.24). Der Fluss einer Kante (i, j, t, τ) im zeitlichen Akquiseplan wird mit $x_{(i, j, t, \tau)}$ bezeichnet. Zunächst werden die Größen des zweiten Optimierungsproblems in Def. 4.21 auf diese Flüsse abgebildet und die Nebenbedingungen als *Fluss-Erhaltungs-Nebenbedingungen* beschrieben:

1. Die Anzahl der Kunden, die zum Zeitpunkt τ dem Akquiseplan AP zugeführt werden – vgl. Def. 4.21, (4.86) – ist:

$$x_{(0, 1, \tau, \tau)} = z_{\tau}(1, \tau) = c_{\tau}$$

für $\tau = 0, 1, \dots, t_{\text{in}}$.

2. Die Anzahl der Kunden, die zum Zeitpunkt τ dem Akquiseplan AP zugeführt werden und im Zeitraum $(t-1, t]$ vom Knoten A_j zum Knoten A_j wechseln, ist:

$$x_{(i, j, t, \tau)} = \Delta z_{\tau}(i, j, t).$$

3. In allen Knoten des zeitlichen Akquiseplans AP^T müssen die *Fluss-Erhaltungs-Nebenbedingungen* eingehalten werden, das heißt für $A_{(j, t, \tau)} \in \underline{A}^T$ mit $j \neq 0, 1, N-1, N+1$:

$$\sum_{(k, t+1, \tau) \in \text{Nachfr}(j, t, \tau)} x_{(j, k, t+1, \tau)} - \sum_{(i, t-1, \tau) \in \text{Vorg}(j, t, \tau)} x_{(i, j, t, \tau)} = 0, \quad (4.107)$$

und für $A_{(0, -1, -1)} \in \underline{A}^T$:

$$\sum_{(k, \tau+1, \tau) \in \text{Nachfr}(1, \tau, \tau)} x_{(1, k, \tau+1, \tau)} - x_{(0, 1, \tau, \tau)} = 0, \quad (4.108)$$

und für $A_{(N-1, t, \tau)}$ mit $t = 0, 1, \dots, t_{\text{end}}$ und $\tau \in T_{(N-1, t)}$:

$$x_{(N-1, N+1, t, \tau)} - \sum_{(i, t-1, \tau) \in \text{Vorg}(N-1, t, \tau)} x_{(i, N-1, t, \tau)} = 0, \quad (4.109)$$

und für $A_{(N, t, \tau)}$ mit $t = 0, 1, \dots, t_{\text{end}}$ und $\tau \in T_{(N, t)}$:

$$x_{(N, N+1, t, \tau)} - \sum_{(i, t-1, \tau) \in \text{Vorg}(N, t, \tau)} x_{(i, N, t, \tau)} = 0, \quad (4.110)$$

und für $A_{(0, -1, -1)}$:

$$x_{(0, 1, \tau, \tau)} - x_{(N+1, 0, -1, -1)} = 0, \quad (4.111)$$

und für $A_{(N+1, -1, -1)}$:

$$\begin{aligned}
& x_{(N+1, 0, -1, -1)} - \sum_{\tau=0, \dots, t_{in}} \left[\sum_{t: \tau \in T_{(N-1, t)}} x_{(N-1, N+1, t, \tau)} \right] \\
& - \sum_{\tau=0, \dots, t_{in}} \left[\sum_{t: \tau \in T_{(N, t)}} x_{(N, N+1, t, \tau)} \right] = 0.
\end{aligned} \tag{4.112}$$

Dies entspricht in Def. 4.21 den Gleichungen (4.85), (4.87), (4.88) und (4.89).

4. Für die neuen Kunden in einem Knoten $A_j \in AP$ gilt zu jedem Zeitpunkt $t = 0, 1, \dots, t_{end}$ die Knoten-Kapazitätsbeschränkung (4.91). Im zeitlichen Akquiseplan AP^T gilt dann

$$\sum_{\tau \in T_{(j, t)}} \sum_{(i, t-1, \tau) \in \text{Vorg}_T(j, t, \tau)} x_{(i, j, t, \tau)} \leq \mu_{l_j}^{-1}(\delta_l) \tag{4.113}$$

für alle $j = 2, \dots, N-2$ und $t = 0, 1, \dots, t_{end}$. Die Knoten A_1, A_{N-1}, A_N besitzen keine Kapazitätsbeschränkungen. In diesen Knoten ist $\mu_{l_j}^{-1} \equiv 1$ für $j = 1, N-1, N$, wäre $\mu_{l_j}^{-1}(\delta_l) = \infty$ für alle $\delta_l \in (0, 1]$.

5. Die Aufteilung von bearbeiteten Kunden in einem Knoten eines Akquiseplans AP auf die Folgeknoten gilt (4.90). Diese Nebenbedingungen stellen sich für den zeitlichen Akquiseplan AP^T wie folgt dar:

$$\left[\xi_{jk}(t+1-\tau) \cdot \sum_{(i, t-1, \tau) \in \text{Vorg}_T(j, t, \tau)} x_{(i, j, t, \tau)} \right] - 1 \leq x_{(j, k, t+1, \tau)}$$

bzw.

$$x_{(j, k, t+1, \tau)} \leq \left[\xi_{jk}(t+1-\tau) \cdot \sum_{(i, t-1, \tau) \in \text{Vorg}_T(j, t, \tau)} x_{(i, j, t, \tau)} \right] + 1 \tag{4.114}$$

für alle $(j, k, t+1, \tau) \in \underline{R}^T$ mit $(j, k) \neq (0, 1), (N-1, N+1), (N, N+1)$.

Die Nebenbedingungen in Punkt 5. entsprechen offensichtlich nicht der Form eines maximaler-Fluss-Problems, da der Fluss zwar bei dieser Problemgattung in jeder Kante bestimmten Schranken unterworfen ist, aber eine (deterministische oder stochastische) Vorgabe der Aufteilung des Flusses an den Knoten nicht vorkommt. Die Behandlung der stochastischen Nebenbedingungen wird zunächst zurückgestellt und im Abschnitt 4.3.4.4 wieder aufgenommen. Die Nebenbedingungen in Punkt 4. entsprechen ebenfalls nicht einem maximaler-Fluss-Problem. Der Unterschied besteht darin, dass Kapazitätsschranken bei einem maximaler-Fluss-Problem nur auf einzelne Kantenflüsse wirken (Glg. 4.96), während sie im vorliegenden Problem zugleich auf mehrere Kantenflüsse wirken können.

Für jeden Zuführungszeitpunkt $\tau = 0, 1, \dots, t_{in}$ besitzt ein zeitlicher Akquise-Plan AP^T einen induzierten Untergraphen (vgl. Anmerkung 3. zur Def. 4.26). Werden diese Untergraphen so überlagert,

- dass die Knoten $A_{(i, t, \tau)}$ für jedes $i = 2, \dots, N-2$ und für jedes $t = 0, \dots, t_{\text{end}}$ zusammengefasst werden:

$$A_{(i, t)} := \{A_{(i, t, \tau)} : \tau \in T_{(i, t)}\}, \quad (4.115)$$

- und werden die Quelle A_0 und die Senke A_{N+1} , die die Untergraphen „zusammenhalten“ weggelassen,

dann ist das Ergebnis ein Graph AP^* mit folgenden Eigenschaften:

- Der Graph besitzt für jeden Zuführungszeitpunkt $\tau = 0, 1, \dots, t_{\text{in}}$ eine Quelle $A_{(1, \tau, \tau)}$ und zwei Senken $A_{(N-1, -1, -1)}$ und $A_{(N, -1, -1)}$.
- Im Graphen überlagern sich $t_{\text{in}} + 1$ Flüsse – jeweils von der Quelle $A_{(1, \tau, \tau)}$ zu den beiden Senken $A_{(N-1, -1, -1)}$ und $A_{(N, -1, -1)}$.
- In den Knoten $A_{(i, t)}$ wirken auf die sich überlagernden Flüsse die Kapazitätsbeschränkungen (4.113) für $i = 2, \dots, N-2$, $t = 0, \dots, t_{\text{end}}$.

Diese Eigenschaften zeigen, dass das zweite Optimierungsproblem (wenn von den stochastischen Nebenbedingungen abgesehen wird) ein *Mehrgüterfluss-Problem* („multicommodity flow problem“) [118] ist. Die darin vorkommenden Kapazitätsbeschränkungen (4.113) werden als Bündel-Nebenbedingungen („bundle constraints“) bezeichnet. Ihr Einfluss auf die Lösung des Problems wird im nächsten Abschnitt diskutiert.

4.3.4.3 Diskussion der Bündel-Nebenbedingungen

In diesem Abschnitt werden die stochastischen Nebenbedingungen im zweiten Optimierungsproblem vernachlässigt und die Bündel-Nebenbedingungen unter dem Gesichtspunkt der Ganzzahligkeit der Lösung des zweiten Optimierungsproblems diskutiert. Zunächst sind zwei Definitionen und die Sätze 4.10 und 4.11 aus [118] und Satz 4.12 aus [120] einzuführen.

Definition 4.28: *Knoten-Kanten-Inzidenzmatrix eines (zeitlichen) Akquiseplans.*

Für jeden Akquiseplan $A = (\underline{A}, \underline{R})$ (zeitlichen Akquiseplan $A^T = (\underline{A}^T, \underline{R}^T)$) mit der Knotenmenge $\{A_1, \dots, A_N\}$ und der Kantenmenge $\{r_1, \dots, r_M\}$ ¹¹⁾ sei seine *Knoten-Kanten-Inzidenzmatrix* eine Matrix $E = (e_{ij}) \in \{-1, 0, 1\}^{N \times M}$ mit den Eigenschaften:

11. Eine formale Unterscheidung zwischen den Knoten und Kanten des Akquiseplans einerseits und des zeitlichen Akquiseplans andererseits, verbunden mit der spezifischen Indizierung von Knoten und Kanten eines zeitlichen Akquiseplans, ist hier nicht notwendig.

$$e_{ij} := \begin{cases} 1 & \text{für } \exists k \in \text{Nachf}(i) : r_j = (A_i, A_k) \\ -1 & \text{für } \exists k \in \text{Vorg}(i) : r_j = (A_k, A_i) \\ 0 & \text{sonst} \end{cases}$$

für $i = 1, \dots, N, j = 1, \dots, M$. □

Definition 4.29: *Total unimodular.*

Eine Matrix $A \in \mathbb{R}^{m \times n}$ ist *total unimodular* genau dann, wenn die Determinanten aller quadratischen Untermatrizen von A entweder 0, 1 oder -1 sind. □

Satz 4.10: Die Knoten-Kanten-Inzidenzmatrix E eines (zeitlichen) Akquiseplans ist total unimodular [a. a. O.]. □

Satz 4.11: Sei A eine Matrix aus $\mathbb{N}^{m \times n}$ und $K(b)$ die Menge der zulässigen Lösungen von

$$Ax \leq b, x \geq 0 \quad (4.116)$$

mit $x \in \mathbb{R}^n, b \in \mathbb{R}^m$. Dann sind die beiden folgenden Aussagen äquivalent:

1. A ist total unimodular.
 2. Für jeden ganzzahligen Vektor b mit $K(b) \neq \emptyset$ sind alle Extrempunkte¹²⁾ von $K(b)$ ganzzahlig.
- [a. a. O.] □

Satz 4.12: Die folgenden Aussagen sind äquivalent:

1. Matrix $A \in \mathbb{R}^{m \times n}$ ist total unimodular.
2. Für jedes $Q \subseteq M = \{1, \dots, m\}$ gibt es Partitionen Q_1 und Q_2 von Q , so dass für jedes $j = 1, \dots, n$ gilt: $\left| \sum_{i \in Q_1} a_{ij} - \sum_{i \in Q_2} a_{ij} \right| \leq 1$.

[a. a. O.] □

Um aus diesen Sätzen Aussagen über die Lösung des zweiten Optimierungsproblems ableiten zu können, werden seine Nebenbedingungen zunächst in die Form (4.116) gebracht. Die

12. x ist ein Extrempunkt von $K(b)$ genau dann, wenn aus der Existenz von x_1, x_2 und $0 < \lambda < 1$ mit $x = \lambda x_1 + (1 - \lambda)x_2$ folgt: $x = x_1 = x_2$ [117].

Matrix A setzt sich zusammen aus einer Knoten-Kanten-Inzidenzmatrix E und einer Matrix für die Bündel-Nebenbedingungen B :

$$A = \begin{pmatrix} B \\ E \end{pmatrix}. \quad (4.117)$$

(Die stochastischen Nebenbedingungen werden hier nicht betrachtet.)

- Die Matrix E ist nach Satz 4.10 total unimodular. In den Gleichungen (4.107) bis (4.112) kommen nur die Koeffizienten -1 und 1 vor. Für jeden Knoten $A_{(i,t,\tau)}$ des zeitlichen Akquiseplans AP^T gibt es genau eine Gleichung in (4.107) – (4.112). Darin sind die Koeffizienten vor den Kantenflüssen $x_{(i,j,t,\tau)}$ gleich 1 , wenn die Kante beim Knoten $A_{(i,t,\tau)}$ beginnt, und gleich -1 , wenn sie dort endet. Alle anderen Kantenflüsse könnten mit Koeffizient 0 eingefügt werden. Ist x der Flussvektor von AP^T (vgl. Def. 4.22), dann ist die kompakte Schreibweise von (4.107) – (4.112):

$$Ex = 0. \quad (4.118)$$

- In (4.113) steht für alle Knoten A_i des Akquiseplans AP mit $i = 2, \dots, N-2$ und für alle Zeitpunkte $t = 0, 1, \dots, t_{\text{end}}$ eine Kapazitätsnebenbedingung. Alle Kantenflüsse, die in $A_{(i,t,\tau)}$ enden, besitzen darin einen Koeffizienten 1 . Jede dieser Gleichungen kann um die darin nicht vorkommenden Kantenflüsse erweitert werden, indem diese mit Koeffizient 0 eingefügt werden. Mit dem Flussvektor x erhält (4.113) dann die Gestalt:

$$Bx \leq \mu_1(\delta_1), \quad (4.119)$$

wobei $\mu_1(\delta_1)$ ein Vektor mit Elementen $\mu_{1_i}(\delta_1)$, $i = 2, \dots, N-2$ ist. Somit ist die Zusammenfassung der Matrizen E und B zu einer Matrix A gemäß Darstellung (4.117) möglich, und da alle Kantenflüsse nicht negativ sind, können (4.118) und (4.119) in der Form (4.116) zusammengefasst werden:

$$Ax \leq b, x \geq 0.$$

Aus dem oben Gesagten folgt $A \in \{-1, 0, 1\}^{m \times n}$, wobei $m = |\underline{A}^T| + (|\underline{A}| - m^*)(t_{\text{end}} + 1)$, $n = |\underline{R}^T|$ und m^* bezeichnet die Anzahl der Knoten des Akquiseplans AP , für die es keine Kapazitätsbeschränkungen gibt. Im vorliegenden Fall ist $m^* = 3$ (Knoten $A_1, A_{N-1}, A_N \in \underline{A}$).

Nunmehr ist zu klären, ob Matrix A total unimodular ist. Wäre dies der Fall, dann würde mit Satz 4.11 folgen, dass alle Extrempunkte der Menge der zulässigen Lösungen $K(b) := \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ für jeden ganzzahligen Vektor b ganzzahlig wären, und somit hätte das Mehrgüterfluss-Problem mit Bündel-Nebenbedingung eine ganzzahlige Lösung, die mit einem Verfahren für reellwertige LP berechnet werden könnte. Aus der Litera-

Es ist bekannt, dass Bündel-Nebenbedingungen im Allgemeinen diese schöne Eigenschaft zerstören [118]. Über die Verhältnisse im vorliegenden Fall gibt der folgende Satz Auskunft:

Satz 4.13: Gegeben sei das zweite Optimierungsproblem gemäß Def. 4.21 (für einen Akquise-Plan AP) in der Gestalt eines Mehrgüterfluss-Problems (für den zeitlichen Akquise-Plan AP^T zu AP) ohne die stochastischen Nebenbedingungen (4.90):

$$x_{(N+1, 0, -1, -1)} \stackrel{!}{=} \max$$

unter den Nebenbedingungen

$$Ax \leq b \text{ mit } A \in \mathbb{R}^{m \times n} \text{ und } x \in \mathbb{R}_+^n.$$

Dann gilt: Für $b \in \mathbb{Z}^m$ besitzt das Optimierungsproblem nur ganzzahlige Lösungen $x \in \mathbb{Z}_+^n$. □

Beweis: Die Elemente des Vektors b gehören entweder zu Knoten-Bilanzgleichungen und sind dann null, oder sie gehören zu Bündel-Nebenbedingungen, also Kapazitäts-Nebenbedingungen und sind dann positiv ganzzahlig. Die Menge der zulässigen Punkte $K(b) := \{x : Ax \leq b, x \in \mathbb{R}_+^n\}$ enthält also zumindest den Nullvektor und ist damit nicht leer. Wenn gezeigt werden kann, dass die Matrix A total unimodular ist, dann folgt die Behauptung aus Satz 4.11.

Nachfolgend wird mit Hilfe von Satz 4.12 gezeigt, dass die Matrix A total unimodular ist. Gemäß (4.117) besteht A aus der Knoten-Kanten-Inzidenzmatrix des zeitlichen Akquise-Plans E und aus der Matrix der Bündel-Nebenbedingungen B . *Im Folgenden wird nur der zeitliche Akquise-Plan AP^T betrachtet.* Die Anzahl seiner Knoten und Kanten wird wie folgt bezeichnet: $M^T := |\underline{A}^T|$, $N^T := |\underline{R}^T|$. Zur Vereinfachung werden die Knoten einfach indiziert: $\underline{A}^T = \{A_1, \dots, A_{M^T}\}$. Die solchermaßen indizierten Knoten sind nicht mit Knoten eines Akquiseplans zu verwechseln! Die Tupel-Indizierung wird im Folgenden nur gebraucht, wenn die einfache Indizierung nicht ausreicht. Weiterhin, werden die Kanten des zeitlichen Akquiseplans vereinfacht bezeichnet: $\underline{R}^T := \{r_1, \dots, r_{N^T}\}$. Da sowohl die Knotenmenge als auch die Kantenmenge eines zeitlichen Akquiseplans endlich sind, ist diese vereinfachte Indizierung möglich. Die Menge I enthalte die einfachen Knotenindizes, die Menge J die einfachen Kantenindizes des zeitlichen Akquiseplans AP^T :

$$I := \{1, \dots, M^T\}, J := \{1, \dots, N^T\}.$$

Bezeichne $r_j^{(0)} \in I$ den einfachen Index des Anfangsknotens und $r_j^{(1)} \in I$ den einfachen Index des Endknotens der Kante $r_j \in \underline{R}^T$.

Jetzt sind folgende Aussagen möglich:

a) Die *Inzidenzmatrix* besitzt entsprechend Def. 4.28 eine Zeile pro Knoten und eine Spalte pro Kante. Weil jede Kante genau zwei Knoten verbindet, steht in jeder Spalte *genau einmal* 1 und *einmal* -1 . Damit nehmen die Summen $\sum_{i \in Q} a_{ij}$ über die Knoten A_i , $i \in Q$, $Q \subseteq I$

für $j \in J$ folgende Werte an:

1. Falls $Q = I$, dann ist: $\sum_{i \in Q} a_{ij} = 0$.
2. Falls $Q \subset I$ und $[r_j^{(0)} \notin Q] \wedge [r_j^{(1)} \notin Q]$, dann ist $\sum_{i \in Q} a_{ij} = 0$.
3. Falls $Q \subset I$ und $[r_j^{(0)} \in Q] \wedge [r_j^{(1)} \notin Q]$, dann ist $\sum_{i \in Q} a_{ij} = 1$.
4. Falls $Q \subset I$ und $[r_j^{(0)} \in Q] \wedge [r_j^{(1)} \in Q]$, dann ist $\sum_{i \in Q} a_{ij} = 0$.
5. Falls $Q \subset I$ und $[r_j^{(0)} \notin Q] \wedge [r_j^{(1)} \in Q]$, dann ist $\sum_{i \in Q} a_{ij} = -1$.

b) Die Matrix der Bündel-Nebenbedingungen B besitzt eine Zeile pro *Knotenklasse*

$$A_{(i, t)} := \{A_{(i, t, \tau)} \in \underline{A}^T : \tau \in T_{(i, t)}\}, \quad (4.120)$$

$i = 2, \dots, N-2$, $t = 0, \dots, t_{\text{end}}$ (siehe (4.115)) und eine Spalte pro Kante r_j , $j \in J$. In der Zeile zur Knotenklasse $A_{(i, t)}$ steht für jede Kante, die in einem der Knoten $A_{(i, t, \tau)} \in A_{(i, t)}$ mündet, in einer Spalte eine 1 . Da eine Kante nur in einem Knoten enden kann, steht in jeder Spalte von B *höchstens einmal* 1 . Sei M^* die Anzahl der Knotenklassen (4.120). Dann ist $M^T + M^*$ die Zeilenanzahl der Matrix A .

Seien $Q_{E, B} := \{1, 2, \dots, M^T, M^T + 1, \dots, M^T + M^*\}$ die Menge der Zeilenindizes von A und o. B. d. A. $Q_E := I$ die Menge der Zeilenindizes von E und $Q_B := Q_{E, B} \setminus I$ die Menge der Zeilenindizes von B . Sei Q eine beliebige Teilmenge von $Q_{E, B}$: $Q \subseteq Q_{E, B}$ und sei $Q = \tilde{Q}_1 \cup \tilde{Q}_2$ eine *vorläufige* Partitionierung von Q derart, dass $\tilde{Q}_1 = Q \cap Q_E$ und $\tilde{Q}_2 = Q \cap Q_B$. Ohne Beschränkung der Allgemeinheit sei $\tilde{Q}_2 = \{i_1, i_2, \dots, i_{|\tilde{Q}_2|}\}$.

Wie unter a) gezeigt wurde gilt $\sum_{i \in Q_1} a_{ij} \in \{-1, 0, 1\}$. Damit ist folgende Partitionierung von J möglich: $J = J_{-1} \cup J_0 \cup J_1$, wobei

$$J_{-1} := \left\{ j \in J : \sum_{i \in Q_1} a_{ij} = -1 \right\},$$

$$J_0 := \left\{ j \in J : \sum_{i \in Q_1} a_{ij} = 0 \right\},$$

$$J_1 := \left\{ j \in J : \sum_{i \in Q_1} a_{ij} = 1 \right\}.$$

Nun wird in $|\tilde{Q}_2|$ Schritten, ausgehend von

- der vorläufigen Partitionierung $\tilde{Q}_1 = Q \cap Q_E$
- und für $Q_1^{(0)} := \tilde{Q}_1$ und $Q_2^{(0)} := \emptyset$
- und $J_{-1}^{(0)} := J_{-1}$, $J_0^{(0)} := J_0$, $J_1^{(0)} := J_1$

eine Partitionierung $Q = Q_1 \cup Q_2$ aufgebaut, die die Eigenschaft der Partitionierung in Satz 4.12, Teil 2 besitzt.

Führe für $k = 1, \dots, |\tilde{Q}_2|$ **durch**:

Wenn $J_{-1}^{(k-1)} = \emptyset$, **dann** füge i_k zur Menge $Q_2^{(k-1)}$ hinzu:

$$Q_1^{(k)} := Q_1^{(k-1)}, \quad Q_2^{(k)} := Q_2^{(k-1)} \cup \{i_k\};$$

$$J_{-1}^{(k)} := J_{-1}^{(k-1)}, \quad J_0^{(k)} := J_0^{(k-1)}, \quad J_1^{(k)} := J_1^{(k-1)}.$$

Resultat: Da $Q_1^{(k)}$ mit $Q_1^{(k-1)}$ übereinstimmt gilt $\sum_{i \in Q_1^{(k)}} a_{ij} \in \{0, 1\}$ für alle $j \in J$.

Da Zeile i_k nur Nullen und Einsen enthält und in jeder Spalte von B höchstens einmal 1 steht ist $\sum_{i \in Q_2^{(k)}} a_{ij} \in \{0, 1\}$ (vgl. oben b). Die Partitionen von J bleiben unverändert:

$$J = J_{-1}^{(k)} \cup J_0^{(k)} \cup J_1^{(k)}.$$

Sonst (das heißt $J_{-1}^{(k-1)} \neq \emptyset$) füge i_k zur Menge $Q_1^{(k-1)}$ hinzu:

$$Q_1^{(k)} := Q_1^{(k-1)} \cup \{i_k\}, \quad Q_2^{(k)} := Q_2^{(k-1)}.$$

Ohne Beschränkung der Allgemeinheit gehöre die Zeile i_k zur Knotenklasse $A_{(i, t)}$ (vgl. (4.120)). Dann gilt:

$$\text{Für alle } j \in J: A_{I_j^{(1)}} \in A_{(i,t)} \text{ ist } a_{i,j} = 1 \quad (4.121)$$

(vgl. oben b). Seien die Indizes dieser Spalten in der Menge $J(i_k) \subseteq J$ zusammengefasst:

$$J(i_k) := \{j \in J : a_{i_k,j} = 1\}.$$

Jetzt sind für jedes $j \in J(i_k)$ drei Fälle zu unterscheiden:

1. $j \in J_{-1}^{(k-1)}$: Dann ist $\sum_{i \in Q_1^{(k)}} a_{ij} = 0$ und somit

$$J_{-1}^{(k)} := J_{-1}^{(k-1)} \setminus \{j\}, J_0^{(k)} := J_0^{(k-1)} \cup \{j\}, J_1^{(k)} := J_1^{(k-1)}.$$

2. $j \in J_0^{(k-1)}$: Dann ist $\sum_{i \in Q_1^{(k)}} a_{ij} = 1$ und somit

$$J_{-1}^{(k)} := J_{-1}^{(k-1)}, J_0^{(k)} := J_0^{(k-1)} \setminus \{j\}, J_1^{(k)} := J_1^{(k-1)} \cup \{j\}.$$

3. $j \in J_1^{(k-1)}$: Das bedeutet $A_{I_j^{(0)}} \in A_{(i,t)}$, was im Widerspruch zu (4.121) steht und deshalb nicht möglich ist.

Resultat: Entweder treten für $Q_1^{(k)}$ die Zeilensummen 0, 1 und –1 auf:

$$\sum_{i \in Q_1^{(k)}} a_{ij} \in \{-1, 0, 1\}. \quad (4.122)$$

Oder die Spalten mit Zeilensumme gleich –1 verschwinden wegen (), so dass

$$\sum_{i \in Q_1^{(k)}} a_{ij} \in \{0, 1\}. \quad (4.123)$$

Da $Q_2^{(k)} = Q_2^{(k-1)} = \emptyset$ übereinstimmt gilt $\sum_{i \in Q_2^{(k)}} a_{ij} = 0$ für alle $j \in J$.

Wegen () und () gilt für die Partitionierung von J : $J = J_{-1}^{(k)} \cup J_0^{(k)} \cup J_1^{(k)}$.

Setze (für $k = |\tilde{Q}_2|$) $Q_1 := Q_1^{(k)}$ und $Q_2 := Q_2^{(k)}$.

Resultat: Nun ist

- entweder $\sum_{i \in Q_1} a_{ij} \in \{0, 1\}$ und $\sum_{i \in Q_2} a_{ij} \in \{0, 1\}$
- oder $\sum_{i \in Q_1} a_{ij} \in \{0, 1, -1\}$ und $\sum_{i \in Q_2} a_{ij} = 0$.

In beiden Fällen folgt: $|\sum_{i \in Q_1} a_{ij} - \sum_{i \in Q_2} a_{ij}| \leq 1$ für alle $j \in J$. ■

Zusammenfassend kann festgehalten werden: Das zweite Optimierungsproblem ohne stochastische Nebenbedingungen ist ein ganzzahliges lineares Optimierungsproblem. Aufgrund seiner

Struktur als Mehrgüterfluss-Problem und den Besonderheiten der Bündel-Nebenbedingungen besitzt das durch Relaxation der Ganzzahligkeitsbedingungen erhaltene reellwertige lineare Optimierungsproblem nur ganzzahlige Lösungen.

Im folgenden Abschnitt wird untersucht, wie das zweite Optimierungsproblem *mit* stochastischen Nebenbedingungen gelöst werden kann.

4.3.4.4 Szenarienbasiertes deterministisches Äquivalent

Die stochastischen Nebenbedingungen beschreiben die Aufteilung der in den Knoten ankommenden Kundenströme auf die nachfolgenden Knoten. Sie folgen dem in Abschnitt 4.3.2 eingeführten *stochastischen Prozess der Kundenverteilung*. Die Koeffizienten der Nebenbedingungen (4.114) sind gebrochen und die Lösung des reellwertigen Optimierungsproblems z_{LP} wird in der Regel nicht ganzzahlig sein. Hier werden zwei Möglichkeiten zur Berücksichtigung der stochastischen Nebenbedingungen aufgezeigt. Die erste, in diesem Abschnitt behandelte Möglichkeit, erzeugt für den stochastischen Prozess der Kundenverteilung *Szenarien* (siehe Def. 4.18). Jedes Szenario ergibt ein deterministisches Äquivalent des stochastischen zweiten Optimierungsproblems gemäß Def. 4.21. Für jedes Szenario ist ein lineares Optimierungsproblem zu lösen. Die Lösungen für eine gegebene Menge Szenarien werden gemittelt. Die zweite im Abschnitt 4.3.4.5 behandelte Möglichkeit verlagert die stochastischen Nebenbedingungen mit Hilfe des *erweiterten Lagrange-Verfahrens*¹³⁾ in die Zielfunktion des Optimierungsproblems. Dies erlaubt die gleichzeitige Berücksichtigung mehrerer Szenarien, so dass nur ein Optimierungsproblem für eine gegebene Menge Szenarien gelöst werden muss, dessen Zielfunktion aber quadratisch ist.

Szenariobasierte Optimierung

Eine ausführliche Behandlung der Grundlagen der szenariobasierten Optimierung würde über den Rahmen dieser Arbeit hinausgehen. Der Übersichtsartikel [40] bezieht sich vor allem auf Probleme der mehrstufigen stochastischen Optimierung. Zu ihrer Lösung werden Mengen von Szenarien benutzt, die sich als Szenariobaum beschreiben lassen. Szenarien können empirischen Ursprungs sein, insbesondere, wenn die Wahrscheinlichkeitsverteilung des stochastischen Prozesses nicht bekannt ist (siehe z. B. [114]), oder für eine gegebene Wahrscheinlichkeitsverteilung berechnet werden. Zur Reduktion der Komplexität der solchermaßen beschriebenen Optimierungsprobleme wird versucht, den Szenariobaum so zu vereinfachen, dass sich die Qualität der mit dem reduzierten Szenariobaum berechneten Lösung nur um ein bestimmtes Maß ändert [42]. Dabei interessieren besonders folgende Fragen:

- Wie können die Szenarien so reduziert werden, dass die Zunahme des Approximationsfehlers einen gegebenen Wert nicht überschreitet?
- Wie stark nimmt der Approximationsfehler durch die Reduktion der Szenarien auf eine gegebene Zahl von Szenarien zu?

13. Grundlage des *erweiterten Lagrange-Verfahrens* ist das *Verfahren der Multiplikatoren-Straffunktion* aus der nichtlinearen Optimierung [124]. Dessen Hauptmittel ist eine Hilfsfunktion, die aus der Zielfunktion und den „unbequemen Nebenbedingungen“ gebildet wird, die in die Zielfunktion überführt werden. Sie wird als *erweiterte Lagrange-Funktion* („augmented Lagrangian function“) bezeichnet.

Eine detaillierte Untersuchung zur Szenarienreduktion findet sich in [139]. Industrielle Anwendungen der szenariobasierten stochastischen Optimierung finden sich unter anderem im Energiesektor [67]. Erste Ansätze in der Ertragssteuerung von Fluggesellschaften auf Flugnetzebene, an deren Entwicklung der Verfasser beteiligt ist, sind in [114] beschrieben.

In der vorliegenden Arbeit werden Szenarien auf Grundlage einer gegebenen Wahrscheinlichkeitsverteilung berechnet.

Lösung für Szenarien in den Nebenbedingungen

Sei z_{LP} das zweite Optimierungsproblem ohne stochastische Nebenbedingungen:

$$z_{LP} := \max\{cx : Ax \leq b, x \in \mathbb{R}_0^+\}, \quad (4.124)$$

$z_{LP}(\Xi)$ das zweite Optimierungsproblem mit stochastischen Nebenbedingungen:

$$z_{LP}(\Xi) := \max\{cx : Ax \leq b, B(\Xi)x \leq d, x \in \mathbb{R}_0^+\}$$

und $z_{LP}(X)$ das zweite Optimierungsproblem mit einer Realisierung der stochastischen Nebenbedingungen für ein *Szenario der Kundenverteilung* X :

$$z_{LP}(X) := \max\{cx : Ax \leq b, B(X)x \leq d, x \in \mathbb{R}_0^+\}. \quad (4.125)$$

Aus Abschnitt 4.3.4.3 ist bekannt, dass z_{LP} eine ganzzahlige Lösung z_{LP}^* besitzt. Da die rechte Seite d nur nicht negative Elemente enthält, besitzt $z_{LP}(X)$ zumindest die triviale Lösung $x = 0$. Sei S die Menge aller Szenarien. Das Optimierungsproblem kann dann wie folgt gelöst werden [88]:

1. Löse das Optimierungsproblem

$$z_{LP}(X) := \max\{cx : Ax \leq b, B(X)x \leq d, x \in \mathbb{R}_0^+\} \quad (4.126)$$

für alle $X \in S$.

2. Berechne einen Mittelwert der Lösungen $z_{LP}^*(X)$, $X \in S$:

$$\bar{z}_{LP}^* := \sum_{X \in S} W\{X\} z_{LP}^*(X). \quad (4.127)$$

Wegen $\Xi(t) := (\xi_{ij}(t))_{i,j} \in [0, 1]^{N \times N}$ kann Ξ unendlich viele Werte annehmen und S umfasst unendlich viele Szenarien. Für praktische Berechnungen wird die Lösung (4.127) approximiert, indem (4.126) für endlich viele Szenarien X^ν , $\nu = 1, \dots, N$ gelöst und aus diesen Lösungen ein Mittelwert berechnet wird:

1. Löse das Optimierungsproblem

$$z_{LP}(X) := \max\{cx : Ax \leq b, B(X)x \leq d, x \in \mathbb{R}_0^+\}$$

für alle $X \in S_N$ mit $S_N = \{X^1, X^2, \dots, X^N\}$.

2. Berechne einen Mittelwert der Lösungen $z_{LP}^*(X^\nu)$, $\nu = 1, \dots, N$:

$$\bar{z}_{LP}^*(S_N) := \frac{\sum_{X \in S_N} W\{X\} z_{LP}^*(X)}{\sum_{X \in S_N} W\{X\}}. \quad (4.128)$$

Wenn die Szenarien durch Monte-Carlo-Simulation gewonnen werden (siehe Abschnitt 4.3.5) und gleich wahrscheinlich sind, dann ist (4.128) einfacher:

$$\bar{z}_{LP}^*(S_N) := \frac{1}{N} \sum_{\nu=1}^N z_{LP}^*(X^\nu). \quad (4.129)$$

Anmerkungen

1. Die Lösung in (4.127) ist die endliche Folge $c_\tau^*(S)$, $\tau = 0, 1, \dots, t_{in}$ der Anzahl der Kunden, die dem Akquiseplan zu den Zeitpunkten $0, 1, \dots, t_{in}$ zugeführt werden, das heißt

$$\bar{c}_\tau^*(S) := \sum_{X \in S} W\{X\} c_\tau^*(X).$$

2. Die Mittelwertbildung ist nicht unproblematisch, denn es ist nicht sichergestellt, dass $\bar{z}_{LP}^*(S)$ für alle Szenarien $X \in S$ zulässig ist. Dieses Problem wird wie folgt „entschärft“: Betrachtet werden Mengen von Szenarien

$$\{X\}_t := \{X' \in S : (\xi'_{ij}(\tau))_{i,j} = (\xi_{ij}(\tau))_{i,j} \text{ für } \tau = 0, \dots, t_{in} \text{ mit } X' = (\xi'_{ij}(\tau))_{i,j} \text{ und } X = (\xi_{ij}(\tau))_{i,j} \text{ und für alle } i, j \in N\}.$$

Das Optimierungsproblem wird nun in jedem Zeitschritt τ für die Szenarienmenge $\{X\}_\tau$ gelöst, die zu allen vorhergehenden Zeitpunkten tatsächlich eingetreten sind, für $\tau = 0, \dots, t_{in}$:

$$\bar{c}_\tau^*(\{X\}_\tau) := \frac{\sum_{X' \in \{X\}_\tau} W\{X'\} c_\tau^*(X')}{\sum_{X' \in \{X\}_\tau} W\{X'\}}. \quad (4.130)$$

3. Der im vorigen Punkt beschriebene Ansatz ist vom „abwarten-und-Teetrinken“-Typ¹⁴⁾, weil er zu jedem Zeitschritt im Zuführungszeitraum die Realisierung des *stochastischen Prozesses*

14. siehe Fußnote 20 auf Seite 90

ses der Kundenverteilung bis zu diesem Zeitpunkt betrachtet. Dem gegenüber ist das ursprüngliche Problem, das den Wert der Optimierungsvariablen für den gesamten Zuführungszeitraum berechnet, vom „jetzt-und-sofort“-Typ¹⁴⁾.

4. Wie weiter oben schon erwähnt, ist die Approximation der Szenarienmengen S in (4.127) und $\{X\}_\tau$ in (4.130) ein eigener Forschungsgegenstand, der im Rahmen dieser Arbeit nicht behandelt werden kann¹⁵⁾. Die Grundidee ist die „kontrollierte“ Aggregation von Szenarien („scenario decomposition“), so dass Aussagen über die Approximationsgenauigkeit möglich sind.
5. Aus (4.125) ist ersichtlich, dass der Zustandsraum des Problems (4.124) durch die stochastischen Nebenbedingungen weiter eingeschränkt wird. Deshalb müssen die Lösungen z_{LP}^* kleiner als der maximale Fluss $z_{LP}(X)$ sein. Somit wird durch Runden der in der Regel reellwertigen Werte des Vektors $z_{LP}(X)$ auf den nächst höheren ganzzahligen Wert der maximal mögliche Fluss nicht überschritten.

4.3.4.5 Erweitertes Lagrange-Verfahren

Die zweite Möglichkeit zur Behandlung der stochastischen Nebenbedingungen verlagert diese mit Hilfe des *erweiterten Lagrange-Verfahrens* in die Zielfunktion des Optimierungsproblems. Dies erlaubt die gleichzeitige Berücksichtigung mehrerer Szenarien, so dass nur ein Optimierungsproblem für eine gegebene Menge Szenarien gelöst werden muss. Da die entstehende Zielfunktion, die *erweiterte Lagrange-Funktion* („augmented Lagrangian function“) quadratisch ist, werden die Optimierungsvariablen in der Regel nicht mehr ganzzahlig sein, wie beim zweiten Optimierungsproblem ohne stochastische Nebenbedingungen. Es würde zu weit führen, hier die zugehörige Theorie wieder zu geben. Sie ist z. B. in [18] beschrieben. Hier sei nur der folgende Satz genannt:

Satz 4.14: Gegeben sei das Optimierungsproblem

$$f(x) \stackrel{!}{=} \min \quad (4.131)$$

unter den Nebenbedingungen

$$h(x) = 0, x \in X$$

für $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$ und $X \subseteq \mathbb{R}^n$ sowie die zugehörige erweiterte Lagrange-Funktion $L_c(x, \lambda): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$:

$$L_c(x, \lambda) = f(x) + \lambda^T h(x) + \frac{c}{2} h(x)^T h(x) \quad (4.132)$$

mit $\lambda \in \mathbb{R}^m$ und $c > 0$.

15. Eine Behandlung des Themas findet sich z. B. in [22, 40, 41, 42, 43, 88, 132, 139].

Wenn f und h stetige Funktionen, X eine abgeschlossene Menge und die Menge $\{x \in X \mid h(x) = 0\}$ nicht leer ist, dann gilt:

Für $k = 0, 1, \dots$ sei $x^{(k)}$ ein globales Minimum des Optimierungsproblems

$$L_{c^{(k)}}(x, \lambda^{(k)}) \stackrel{!}{=} \min \quad (4.133)$$

unter der Nebenbedingung

$$x \in X.$$

Wenn die Folge $\{\lambda^{(k)}\}$ beschränkt ist und $0 < c^{(k)} < c^{(k+1)}$ für alle k und $c^{(k)} \rightarrow \infty$ gilt, dann ist jeder Grenzwert der Folge $\{x^{(k)}\}$ ein globales Minimum des ursprünglichen Optimierungsproblems. \square

Beweis: siehe [18] ■

Bertsekas nennt zwei Mechanismen, durch welche ein Wert von x nahe dem Optimalwert x' bei unbeschränkter Optimierung von (4.132) berechnet wird [18]:

1. Wenn λ nahe dem Vektor der Lagrange-Multiplikatoren λ' liegt, dann befindet sich die Lösung des Optimierungsproblems (4.133) x nahe dem Optimalwert x' , sofern der Parameter c einen bestimmten Schwellenwert überschreitet.
2. Wenn c sehr groß ist, dann sind die Kosten einer unzulässigen Lösung sehr hoch. Deshalb kann erwartet werden, dass die Lösung des Optimierungsproblems (4.133) annähernd zulässig ist, das heißt $L_c(x, \lambda) \approx f(x)$ für annähernd zulässiges x . Daher ist für große Werte von c zu erwarten, dass die Lösung von (4.133) eine gute Näherung von x' ergibt.

Wichtig ist die Feststellung unter Punkt 1., dass c nicht beliebig wachsen muss, um für eine beschränkte Folge $\{\lambda^{(k)}\}$ die Konvergenz von $\{x^{(k)}\}$ gegen x' zu gewährleisten. Dadurch kann eine schlechte Konditionierung des Problems vermieden werden, die für $c \rightarrow \infty$ auftreten würde. Alle in der Literatur angegebenen Algorithmen zur Lösung von (4.133) sehen deshalb zwei Iterationszyklen vor. In einem Iterationszyklus wird der Parameter c so lange vergrößert, bis die Konvergenz $x^{(k)} \rightarrow x'$ gewährleistet ist. In einem zweiten Zyklus wird $\lambda^{(k)}$ so iteriert, dass $\lambda^{(k)} \rightarrow \lambda'$ gilt und dann das Lösen von (4.133) die Lösung des ursprünglichen Problems x' ergibt. Der Alg. 4.6 vereint alle Merkmale der Algorithmen aus [55, 124, 18, 121].

Anmerkungen

1. Im Algorithmus wird anstelle eines Skalars c ein Vektor $c \in \mathbb{R}^m$ verwendet. Dadurch kann c komponentenweise für jede einzelne Nebenbedingung $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$

```

Eingabe  $\lambda^{(0)} \geq 0, c^{(0)} \geq 0, \beta > 1, \gamma > 0, \varepsilon > 0;$            // Parameter initialisieren
Setze  $k := 0, f_{\text{Lsg}} := \text{false}, \|h^{(0)}\| := \infty;$            // Initialisierung
Solange  $f_{\text{Lsg}} = \text{false}$  führe aus {                               // Flag zur Steuerung der Iteration

    Löse das Optimierungsproblem  $x(\lambda^{(k)}) := \min \{ L_{c^{(k)}}(x, \lambda^{(k)}) \mid x \in X \};$ 

    Setze  $x^{(k+1)} := x(\lambda^{(k)});$ 
    Berechne  $h^{(k+1)} := h(x^{(k+1)});$ 
    Wenn  $\|h^{(k)}\| < \varepsilon$  dann                                     // Abbruchkriterium
        Setze  $f_{\text{Lsg}} := \text{true};$ 
    Wenn  $f_{\text{Lsg}} = \text{false}$  dann führe aus {                           // Iterationsschleife
        Für  $i = 1, \dots, m$  führe aus {
            Wenn  $|h_i^{(k+1)}| > \gamma |h_i^{(k)}|$  dann
                Setze  $c_i^{(k+1)} := \beta c_i^{(k)};$ 
            }
        Setze  $C^{(k+1)} := \text{diag}(c_i^{(k+1)});$ 
        Setze  $\lambda^{(k+1)} := \lambda^{(k)} + C^{(k+1)} h^{(k+1)};$ 
        Setze  $k := k + 1;$ 
    }
}
Ausgabe:  $x^{(k)};$ 

```

Algorithmus 4.6: Algorithmus zur Lösung von (4.131) mit dem erweiterten Lagrange-Verfahren.

erhöht werden. Dies erfordert eine andere Formulierung der erweiterten Lagrange-Funktion als in (4.132) angegeben:

$$L_c(x, \lambda) = f(x) + \lambda^T h(x) + \frac{1}{2} h(x)^T C h(x) \quad (4.134)$$

mit $C := \text{diag}(c_1, \dots, c_m) \in \mathbb{R}^{m \times m}$.

2. Für die Initialisierung der Parameter sind in der Literatur folgende Hinweise zu finden:

- $\lambda^{(0)}$ soll möglichst nahe beim Vektor der Lagrange-Multiplikatoren λ^* liegen [18]. Dieser Hinweis ist von geringem praktischen Nutzen.
- Die Parameter β und γ steuern die Vergrößerung des Parameters c . Er soll nicht zu groß sein, damit das Problem nicht schlecht konditioniert wird. Er soll deshalb auch nicht zu schnell vergrößert werden. Andererseits soll die Konvergenz der Iteration nicht durch zu langsames Wachstum dieses Parameters verzögert werden [18]. Als konkrete Werte nennt Bertsekas $\beta \in [5, 10]$ und $\gamma = 0,25$. In [55] und [124] werden $\beta = 10$ und $\gamma = 0,25$ gewählt. In [121] wird c in jedem Iterationsschritt vergrößert, das heißt die Abfrage $|h_i^{(k+1)}| > \gamma |h_i^{(k)}|$ fehlt bei ihm. Und für die Vergrößerung von c findet sich nur die allge-

meine Vorschrift: $c^{(k+1)} \in (c^{(k)}, \infty)$. Nocedal gibt daneben noch den bezüglich der Parametersteuerung „verfeinerten“ Algorithmus mit der Bezeichnung „LANCELOT“ von Conn, Gould und Toint an.

Für die Implementierung des Verfahrens schließt sich der Verfasser der „Mehrheit“ an und wählt $\beta = 10$ und $\gamma = 0,25$. Diese Werte haben sich bei der praktischen Berechnung in Abschnitt 4.4 bewährt.

3. Unterschiede bestehen auch bei der Lösung des Optimierungsproblems. So wird im Algorithmus von Nocedal nur eine Näherungslösung verlangt, die der Bedingung $\|\nabla_x L_{c^{(k)}}(x, \lambda^{(k)})\|_\infty \leq \tau_k$ genügt. Dieses Kriterium wird auch als Abbruchkriterium anstelle von $\|h^{(k)}\| < \varepsilon$ gewählt.

Eine Verfeinerung gegenüber der Wahl eines festen Wertes für τ_k wird in Bertsekas angegeben. Er führt zwei Parameter-Nullfolgen $\varepsilon_1^{(k)}, \varepsilon_2^{(k)} \rightarrow 0$ ein und setzt als Abbruchkriterium: $\|\nabla_x L_{c^{(k)}}(x, \lambda^{(k)})\|_\infty \leq \min\{\varepsilon_1^{(k)}, \varepsilon_2^{(k)} \cdot \|c^{(k+1)}\|_\infty\}$.

Für die Implementierung des Verfahrens hat sich das einfache Abbruchkriterium $\|h^{(k)}\| < \varepsilon$ als ausreichend erwiesen.

4. [18] weist darauf hin, dass sich das erweiterte Lagrange-Verfahren in einer großen Anzahl praktischer Anwendungen als ziemlich zuverlässig erwiesen hat und in der Regel zumindest zu einem lokalen Minimum konvergiert.

Nachdem die theoretischen Grundlagen des erweiterten Lagrange-Verfahrens benannt und ein Algorithmus zu seiner Anwendung vorgestellt ist, bleibt zu prüfen, wie das zweite Optimierungsproblem mit stochastischen Nebenbedingungen (4.114) damit gelöst werden kann. Dazu werden die Voraussetzungen von Satz 4.14 überprüft. Seien zunächst die Kantenflüsse in einem Vektor zusammengefasst:

$$x := (x_{(i,j,t,\tau)})_{x_{(i,j,t,\tau)} \in \mathbb{R}^T}.$$

Sei s ein Vektor noch genauer zu bestimmender Schlupfvariablen, dann ist

$$\tilde{x} := (x, s)$$

der Vektor der Optimierungsvariablen. Die Zielfunktion lautet

$$f(\tilde{x}) := - \sum_{\tau=0}^{t_{in}} x_{(0,1,\tau,\tau)}.$$

Sie ist linear und stetig in den Optimierungsvariablen. Sei $n_1 := |\mathbb{R}^T|$ die Anzahl der Optimierungsvariablen, die Kantenflüssen entsprechen und n_2 die Anzahl der Schlupfvariablen sowie $n = n_1 + n_2$. Sei weiterhin m_1 die Anzahl der Fluss-Erhaltungs-Nebenbedingungen und m_2

die Anzahl der Bündel-Nebenbedingungen. Nun sei die Menge X die Teilmenge des \mathbb{R}_+^n , für die die Fluss-Erhaltungs-Nebenbedingungen und die Bündel-Nebenbedingungen erfüllt sind. Zur Beschreibung von X werden die Funktionen $g^1 : \mathbb{R}_+^n \rightarrow \mathbb{R}^{m_1}$ und $g^2 : \mathbb{R}_+^n \rightarrow \mathbb{R}^{m_2}$ eingeführt sowie $g = (g^1, g^2)$, so dass

$$X := \{\tilde{x} \in \mathbb{R}_+^n : g(\tilde{x}) = 0\}. \quad (4.135)$$

Für jeden Knoten des zeitlichen Akquiseplans $A_{(i,j,\tau)} \in \underline{A}^T$ gibt es eine Fluss-Erhaltungs-Nebenbedingung, die in den Gleichungen (4.107) – (4.112) beschrieben ist. Sie bilden die m_1 Komponenten von g^1 :

$$g_{(i,j,\tau)}^1(\tilde{x}) = 0 \text{ für } A_{(i,j,\tau)} \in \underline{A}^T.$$

Für jeden Knoten des Akquiseplans AP mit einer Kapazitätsbeschränkung, das heißt für $A_j \in \underline{A}$ mit $j = 2, \dots, N-2$ gibt es eine Bündel-Nebenbedingung, die zu jedem Zeitpunkt $t = 0, 1, \dots, t_{\text{end}}$ gilt. Um diese in (4.113) beschriebenen Ungleichungen auf die gewünschte Gleichungsform zu bringen, wird für jeden Knoten A_j und jeden Zeitpunkt t eine Schlupfvariable $s_{(j,t)} \geq 0$ eingeführt:

$$\sum_{\tau \in T_{(j,t)}} \sum_{(i,t-1,\tau) \in \text{Vorg}_T(j,t,\tau)} x_{(i,j,t,\tau)} + s_{(j,t)} - \mu_{I_j}^{-1}(\delta_I) = 0.$$

Sie bilden die m_2 Komponenten von g^2 :

$$g_{(j,t)}^2(\tilde{x}) = 0 \text{ für } A_j \in \underline{A} \text{ und } t = 0, 1, \dots, t_{\text{end}}.$$

Ein $\tilde{x} \in \mathbb{R}_+^n$ befindet sich also genau dann in der Menge (4.135), wenn die Fluss-Erhaltungs-Nebenbedingungen und die Bündel-Nebenbedingungen erfüllt sind. Sei $(\tilde{x}^{(k)})_{k \in \mathbb{N}}$ eine beliebige Folge in X , die gegen den Wert x^* konvergiert. Die Funktion g ist offensichtlich stetig und daher konvergiert die Folge $(g(\tilde{x}^{(k)}))_{k \in \mathbb{N}}$ gegen den Wert $g(x^*)$. Da $g(\tilde{x}^{(k)}) = 0$ für alle $\tilde{x}^{(k)} \in X$, muss der Grenzwert $g(x^*)$ gleich null sein, und damit liegt x^* in X . Somit ist gezeigt, dass X abgeschlossen ist.

Nun wird die Funktion $h : \mathbb{R}_+^n \rightarrow \mathbb{R}^{m_3}$ über die stochastischen Nebenbedingungen (4.114) definiert. Dazu werden diese zunächst durch Einführung von Schlupfvariablen $s_{(j,t,\tau)} \geq 0$ auf Gleichungsform gebracht:

$$\left[\xi_{jk}(t+1-\tau) \cdot \sum_{(i, t-1, \tau) \in \overline{\text{Vorg}_T(j, t, \tau)}} x_{(i, j, t, \tau)} \right] - 1 + s_{(j, t, \tau)} - x_{(j, k, t+1, \tau)} = 0 \quad (4.136)$$

und

$$\left[\xi_{jk}(t+1-\tau) \cdot \sum_{(i, t-1, \tau) \in \overline{\text{Vorg}_T(j, t, \tau)}} x_{(i, j, t, \tau)} \right] + 1 - s'_{(j, t, \tau)} - x_{(j, k, t+1, \tau)} = 0 \quad (4.137)$$

für jede von einem Knoten des zeitlichen Akquiseplans abgehende Kante $(j, k, t+1, \tau) \in \underline{\mathbf{R}}^T$ mit $(j, k) \neq (0, 1), (N-1, N+1), (N, N+1)$.

Sie bilden die m_3 Komponenten von $h : \mathbf{R}_+^n \rightarrow \mathbf{R}^{m_3}$:

$$h_{(j, k, t+1, \tau)}(\tilde{\mathbf{x}}) = 0 \quad (4.138)$$

für $(j, k, t+1, \tau) \in \underline{\mathbf{R}}^T$ mit $(j, k) \neq (0, 1), (N-1, N+1), (N, N+1)$.

Die Funktion $h(\tilde{\mathbf{x}})$ ist linear in den Optimierungsvariablen \mathbf{x} und in den Schlupfvariablen \mathbf{s} und stetig. Es ist $\mathbf{x} \in \mathbf{X}$ und $\mathbf{s} \in \mathbf{R}_+^{n_2}$. Sei

$$\tilde{\mathbf{X}} := \mathbf{X} \times \mathbf{R}_+^{n_2}.$$

Jetzt ist noch zu zeigen, dass $\{\tilde{\mathbf{x}} \in \tilde{\mathbf{X}} \mid h(\tilde{\mathbf{x}}) = 0\} \neq \emptyset$. Offensichtlich gilt $h(0) = 0$ und $0 \in \tilde{\mathbf{X}}$, so dass auch diese Bedingung von Satz 4.14 erfüllt ist. Damit kann die erweiterte Lagrange-Funktion (4.134) für das Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen formuliert werden:

$$L_c(\tilde{\mathbf{x}}, \lambda) = f(\tilde{\mathbf{x}}) + \lambda^T h(\tilde{\mathbf{x}}) + \frac{1}{2} h(\tilde{\mathbf{x}})^T C h(\tilde{\mathbf{x}})$$

für $\tilde{\mathbf{x}} \in \mathbf{R}_+^{n_1} \times \mathbf{R}_+^{n_2}$, $\lambda \in \mathbf{R}_+^{m_3}$, $h : \mathbf{R}_+^n \rightarrow \mathbf{R}^{m_3}$ mit Komponenten (4.136) und (4.137) und $C = \text{diag}(c_1, \dots, c_{m_3})$ mit $c_i > 0$ für $i = 1, \dots, m_3$.

Die Nebenbedingungen (4.138) enthalten Zufallsvariablen. Zur Verdeutlichung wird $h(\tilde{\mathbf{x}}, \Xi)$ geschrieben, wobei $\Xi := \Xi(t) = (\xi_{ij}(t))_{i,j} \in [0, 1]^{N \times N}$ den stochastischen Prozess der Kundenverteilung gemäß Def. 4.18 bezeichnet. Im gleichen Sinne wird für die erweiterte Lagrange-Funktion

$$\begin{aligned}
L_c(\tilde{x}, \lambda, \Xi) &= f(\tilde{x}) + \lambda^T h(\tilde{x}, \Xi) + \frac{1}{2} h(\tilde{x}, \Xi)^T C h(\tilde{x}, \Xi) \\
&= f(\tilde{x}) + \lambda^T h(\tilde{x}, \Xi) + \frac{1}{2} \sum_{i=1}^{m_1+m_2} c_i h^2(\tilde{x}, \Xi)
\end{aligned}$$

geschrieben.

Das Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen lässt sich somit mit Alg. 4.7 für deterministische Äquivalente von $L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, \Xi)$ lösen. Innerhalb der Iteration ist dabei das folgende Optimierungsproblem zu lösen:

$$L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, \Xi) = f(\tilde{x}) + \lambda^{(k)T} h(\tilde{x}, \Xi) + \frac{1}{2} \sum_{i=1}^{m_3} c_i^{(k)} h^2(\tilde{x}, \Xi) \stackrel{!}{=} \min \quad (4.139)$$

unter der Nebenbedingung $\tilde{x} \in \tilde{X}$ gemäß (4.135).

Bestimmung eines deterministischen Äquivalents

Anders als im Abschnitt 4.3.4.4 wird hier das Optimierungsproblem nicht für mehrere Szenarien gelöst und die Lösungen gemittelt, sondern die Lösung bei gleichzeitiger Betrachtung aller Szenarien approximiert. Sei Ω die Menge aller stochastischen Prozesse der Kundenverteilung Ξ . Dann kann die Lösung

$$x(\lambda^{(k)}, \Xi) = \min \left\{ \int_{\Omega} L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, \Xi) P(d\Xi) \mid \tilde{x} \in \tilde{X} \right\}$$

durch N unabhängige Stichproben der Kundenverteilung, X^1, X^2, \dots, X^N angenähert werden [139]:

$$x(\lambda^{(k)}, N) = \min \left\{ \frac{1}{N} \sum_{\nu=1}^N L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, X^\nu) \mid \tilde{x} \in \tilde{X} \right\}. \quad (4.140)$$

$x(\lambda^{(k)}, N)$ bezeichnet eine Näherungslösung bei Stichprobennahme mit N Stichproben. Sie ist freilich nicht eindeutig, da die Stichprobennahme zufällig erfolgt. Birge beleuchtet die Güte der Näherungslösung $x(\lambda^{(k)}, N)$ gegenüber der genauen Lösung $x(\lambda^{(k)}, \Xi)$ unter zwei Gesichtspunkten [22]:

1. Sei $x(N) := (x_1(\lambda^{(k)}, N), \dots, x_r(\lambda^{(k)}, N))$ ein Vektor mit r unabhängigen Stichproben der Kundenverteilung. Dann ist es unter bestimmten Voraussetzungen möglich, eine diskrete Verteilung $u = (u_1, \dots, u_r)$ anzugeben, so dass

$$\sqrt{N}(x(N) - x(\lambda^{(k)}, \Xi)) \rightarrow u \text{ für } N \rightarrow \infty.$$

2. Für Näherungslösungen $x(\lambda^{(k)}, N)$ gilt unter bestimmten Voraussetzungen

$$\sqrt{N}(x(\lambda^{(k)}, N) - x(\lambda^{(k)}, \Xi)) \rightarrow N\left(0, \text{Var}_{L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, \Xi)}(x^0)\right) \text{ für } N \rightarrow \infty,$$

wobei x^0 das eindeutige Minimum des Erwartungswertes von $L_{c^{(k)}}(\tilde{x}, \lambda^{(k)}, \Xi)$ ist.

Beide Aussagen gelten unter Voraussetzungen, deren Überprüfung schwierig ist. Anstatt hier näher darauf einzugehen, sollen im Folgenden Lösungen für unterschiedlich hohe Anzahlen von Szenarien berechnet, betrachtet und verglichen werden.

4.3.5 Generierung von Zufallszahlen

Für die Erzeugung der Szenarien müssen die normiert multinomialverteilten Zufallsvektoren in (4.70) generiert werden:

$$(\xi_{ij}(t))_{j \in \overline{\text{Nachf}(i)}} \sim M_{c, (\Psi_{ij}^t)_{j \in \overline{\text{Nachf}(i)}}}^n.$$

Da die Normierung nur die Division der Realisierungen eines multinomialverteilten Zufallsvektors erfordert, ist zu klären, wie letztere zu erzeugen sind. In der Literatur sind zahlreiche Ansätze zur Erzeugung von Zufallszahlen beschrieben [36, 54, 59], deren Motivation in der Reduktion des für die Berechnung erforderlichen Rechenzeit- und Speicheraufwands liegt. Dabei wird zwischen Zufallszahlen mit Gleichverteilung und solchen mit anderen Verteilungen und zwischen Zufallszahlen mit eindimensionaler und solchen mit mehrdimensionaler Verteilung unterschieden. Die Berechnung mehrdimensional verteilter Zufallsvariablen wird auf die Berechnung eindimensional verteilter Zufallszahlen zurückgeführt. Die Berechnung von Zufallsvariablen mit Gleichverteilung ist die Grundlage für die Berechnung von Zufallszahlen mit anderer Verteilung. Für diese Transformation gibt es verschiedene Grundprinzipien, die in der angeführten Literatur ausführlich beschrieben werden.

Für die Erzeugung multinomialverteilter Zufallsvektoren ist ein Verfahren in drei Schritten üblich¹⁶⁾:

1. Generierung multinomialverteilter Zufallsvektoren aus einvariablen binomialverteilten Zufallszahlen,
2. Generierung einvariabler binomialverteilter Zufallszahlen aus einvariablen gleichverteilten Zufallszahlen,
3. Generierung einvariabler gleichverteilter Zufallszahlen.

16. Die Nummerierung entspricht der Aufrufreihenfolge in einem Rechenprogramm.

Für Schritt 3 stellt die Programmiersprache C nach ISO/IEC-Standard¹⁷⁾ die Funktion `int rand(void)` zur Verfügung, die für eine vorgegebene Initialisierung eine identische Folge von Pseudozufallszahlen zwischen 0 und einem Maximalwert `RAND_MAX` erzeugt. Der Maximalwert beträgt gemäß *Standard-C* mindestens 32767. Sein konkreter Wert hängt vom Übersetzerprogramm, dem Betriebssystem und dem Rechner ab¹⁸⁾ [90, 133]. Die Funktion `rand()` gehört auch zum Standard der Programmiersprache C++ [142, 28], die vom Verfasser für die Implementierung des Fallbeispiels im Abschnitt 4.4 verwendet wird.

Der geringe Maximalwert der von `rand()` erzeugten Zufallszahlen ist dem maximalen Produkt aus der Anzahl der Kunden in einem Knoten und der Anzahl der Folgeknoten des Knotens gegenüberzustellen. Für `RAND_MAX = 32767` wären also 1000 Kunden und 30 Folgeknoten unproblematisch. Diese Größenordnung wird im Fallbeispiel im Abschnitt 4.4 nicht überschritten. Für größere Akquisepläne und Kundenzahlen wäre `RAND_MAX = 32767` unzureichend und ein Generator für gleichverteilte Zufallszahlen mit einem größeren Maximalwert zu implementieren. Konkrete Implementierungsvorschläge dazu sind in [133] zu finden, grundsätzliche Verfahren zur Erzeugung gleichverteilter Zufallszahlen in [36, 54, 59]¹⁹⁾.

Das einfachste Verfahren zur Generierung binomialverteilter Zufallszahlen (Schritt 2) ist das Bernoulli- oder Münzwurfverfahren. Seine Grundlage ist folgender Hilfssatz [36]:

Hilfssatz 4.15: Sei X die Anzahl der erfolgreichen Versuche in einer Folge von n Bernoulli-Versuchen²⁰⁾ mit Wahrscheinlichkeit p , das heißt

$$X = \sum_{i=1}^n 1_{I_p(U_i)},$$

wobei U_1, \dots, U_n unabhängig identisch in $[0, 1]$ gleichverteilte Zufallsvariablen sind und $1_{I_p(U_i)}$ die Indikatorfunktion der Menge $I_p(x) := \{x \in \mathbb{R} \mid 0 \leq x < p\}$ ist. Dann gilt:

-
17. Standard-C ist der von der International Organization for Standardization (ISO) und der International Electrotechnical Commission (IEC) festgelegte Standard ISO/IEC 9899:1990 (<http://www.iso.org>).
18. Der geringe Wert von `RAND_MAX` scheint nicht antiquiert zu sein. Auf dem keineswegs antiquierten Rechner des Verfassers (AMD Athlon XP 1700+-Prozessor, Betriebssystem Windows XP, *Microsoft Visual C++ 6.0*) ist `RAND_MAX` jedenfalls nicht größer als der vom Standard vorgeschriebene Mindestwert.
19. Aufgrund der Abhängigkeit der Größe von `RAND_MAX` von Rechner, Betriebssystem und Übersetzerprogramm ist die Funktion `rand()` nur bedingt portabel. Wird die Bedingung `RAND_MAX > 32767` in einer „Umgebung“ erfüllt, so kann die Bedingung in einer anderen Umgebung verletzt sein. Generell empfiehlt sich die Implementierung eines eigenen Zufallszahlengenerators.
20. Sei $(X_i)_{1 \leq i \leq n}$ eine unabhängige Familie identisch verteilter Zufallsvariablen mit $W\{X_i = 1\} =: p$ und $W\{X_i = 0\} =: 1 - p$ für $i = 1, \dots, n$. Solche Familien von X_i beschreiben Versuche, bei denen keine gegenseitige Beeinflussung vorliegt und bei denen jeweils eine der beiden Alternativen („1“) mit von i unabhängiger Wahrscheinlichkeit p und die andere Alternative („0“) mit Wahrscheinlichkeit $1 - p$ auftritt. Man spricht hier von einer Folge von *Bernoulli-Versuchen* [74].
-

$$X \sim B_{n,p}.$$

□

Eine $B_{n,p}$ -verteilte Zufallsvariable kann somit erzeugt werden, indem n -mal hintereinander eine im Intervall $[0, 1]$ gleichverteilte Zufallsvariable U erzeugt wird, z. B. durch Berechnung von $\text{rand}() / (\text{RAND_MAX} + 1.0)$ und gezählt wird, wie oft der Wert von U kleiner als p ist. Die Methode gehört zur Gruppe der inversen Transformationsverfahren und besitzt für

$$n \cdot p < 500 \quad (4.141)$$

eine $O(n)$ -Rechenzeit²¹⁾, die sich mit anderen Verfahren messen kann [59]. Wegen der Symmetrie

$$X \sim B_{n,p} \Leftrightarrow (n - X) \sim B_{n,1-p}$$

gilt stets $p < 0,5$, so dass (4.141) für $n \leq 1000$ eingehalten wird. Das Bernoulli-Verfahren ist somit für das Fallbeispiel im Abschnitt 4.4 ausreichend. Für größere Werte von n werden in [54] die schnelleren Algorithmen *ITR* und *BRUA** beschrieben. *ITR* basiert wie das Bernoulli-Verfahren auf inverser Transformation und besitzt eine $O(np)$ -Rechenzeit. *BRUA** gehört zur Verfahrensgruppe „Verhältnis der gleichverteilten Zufallsvariablen“ („ratio-of-uniforms method“). [54] empfiehlt die Verwendung von *ITR* für $np \leq 10$ und *BRUA** sonst.

Für die Generierung multinomialverteilter Zufallsvektoren aus einvariablen binomialverteilten Zufallszahlen (Schritt 2) wird in der Literatur zumeist ein „bedingte Wahrscheinlichkeit-Verfahren“ („conditional probability method“) beschrieben. Dem liegt folgende Eigenschaft zugrunde [36]:

Hilfssatz 4.16: Gegeben sei der multinomialverteilte Vektor (X_1, \dots, X_m) ,

$$(X_1, \dots, X_m) \sim M_{n, (p_1, \dots, p_m)}.$$

Dann ist X_1 binomialverteilt,

$$X_1 \sim B_{n, p_1},$$

und der Vektor (X_2, \dots, X_m) ist multinomialverteilt,

$$(X_2, \dots, X_m) \sim M_{n-X_1, (q_2, \dots, q_m)},$$

mit $q_j = p_j / (1 - p_1)$ für $j = 2, \dots, m$.

□

21. Die Rechenzeit ist eine $O(f(n))$ -Funktion, wenn sie für fast alle $n \in \mathbb{N}$ betragsmäßig höchstens so groß wie ein konstantes Vielfaches von $f(n)$ ist [3].

Für den darauf basierenden Algorithmus (siehe Alg. 4.7) macht [54] die Aussage, dass die durchschnittliche Anzahl zu generierender binomialverteilter Zufallsvariablen minimal ist, wenn die Wahrscheinlichkeiten p_1, \dots, p_m absteigend geordnet sind:

$$p_1 \geq p_2 \geq \dots \geq p_m.$$

```

Eingabe  $n, m, p_1, \dots, p_m$ ;
Setze  $q := 1, k := n, j := 1$ ;
 $X := (\underbrace{0, \dots, 0}_{m\text{-mal}})$ 
Setze  $\quad \quad \quad$ ;
Solange  $k \cdot (m - j + 1) \neq 0$  führe aus {
    Generiere eine Zufallszahl  $Y \sim B_{k, p_j/q}$ ;
    Setze  $X_j := Y$ ;
    Setze  $k := k - X_j, q := q - p_j, j := j + 1$ ;
}
Ausgabe:  $X$ ;

```

Algorithmus 4.7: Algorithmus zur Erzeugung eines multinomialverteilten Zufallsvektors.

4.4 Fallbeispiel

In den vorigen Abschnitten wurden Modelle für das zweite Optimierungsproblem entwickelt. In diesem Abschnitt wird das Modell des zweiten Optimierungsproblems als Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen anhand eines Fallbeispiels, das auf dem Akquiseplan in Abb. 3.5 beruht, untersucht. Das Modell und seine Lösungsverfahren für das Mehrgüterfluss-Problem mit und ohne stochastische Nebenbedingungen sind in der Programmiersprache C++ programmiert²²⁾. Die Implementierung erfolgt mit der Entwicklungsumgebung *Visual Studio*²³⁾, enthält aber keinen *Microsoft*-spezifischen Quellcode, sondern folgt dem *Standard-C++*²⁴⁾ und nutzt auch Elemente der *C++ STL*²⁵⁾. Zur Lösung der linearen und quadratischen Optimierungsprobleme wird der Standard-Löser *CPLEX* verwendet und über eine *C++-Schnittstelle*²⁶⁾ angesprochen. Für einfache Hilfsfunktionen wird die generische

22. C++ ist eine objektorientierte Sprache, die ab 1980 von Bjarne Stroustrup aus der älteren Programmiersprache C entwickelt wurde und seit 1983 diese Bezeichnung trägt [28].

23. *Visual Studio* ist eine Software-Entwicklungsumgebung der Fa. *Microsoft* (<http://www.microsoft.com>). Verwendet wird die Version 6.0.

24. *Standard-C++* ist der von der International Organization for Standardization (ISO) und der International Electrotechnical Commission (IEC) festgelegte Standard ISO/IEC 14882:2003 (<http://www.iso.org>). *Microsoft Visual C++* ist eine Erweiterung dieses Standards, die unter anderem die *Microsoft Foundation Class Library* (MFC) umfasst. Letztere wird für die vorliegende Arbeit nicht verwendet.

25. Die *Standard Template Library*, *STL* ist Teil des C++-Standards und enthält zahlreiche für die mathematische Modellierung nützliche Template-Klassen, z. B. Vektoren.

26. *CPLEX* ist ein Löser für verschiedene Arten von Optimierungsproblemen der Firma *ILOG* (<http://www.ilog.fr>). Die C++-Schnittstelle ist eine C++-Klassen-Bibliothek mit der Bezeichnung *Concert Technology* [78]. Verwendet wird die *CPLEX*-Version 8.1 sowie die Version 1.3 der *Concert* Klassen-Bibliothek.

Klassenbibliothek *genutil.lib* verwendet, die von *Lufthansa Systems Berlin*²⁷⁾, Bereich *Revenue Management* entwickelt wurde und intern in der Software-Produktion eingesetzt wird. Zunächst wird im Abschnitt 4.4.1 das Computerprogramm beschrieben, danach folgt die Untersuchung des Fallbeispiels im Abschnitt 4.4.2.

4.4.1 Implementierung des Verfahrens

Alle nachfolgend genannten Quell-, Parameter-, Eingabe- und Ausgabe-Dateien können von der Homepage des Verfassers²⁸⁾ heruntergeladen werden.

Für die Untersuchung des Fallbeispiels werden in den folgenden Abschnitten 4.4.2.1 – 4.4.2.3 folgende Optimierungsprobleme betrachtet:

- a) Das zweite Optimierungsproblem ohne stochastische Nebenbedingungen: ein Mehrgüterfluss-Problem mit Bündel-Nebenbedingungen, das eine ganzzahlige Lösung besitzt (siehe Abschnitt 4.3.4.3);
- b) das zweite Optimierungsproblem mit stochastischen Nebenbedingungen, szenarienbasiertem deterministischem Äquivalent und dem in Abschnitt 4.3.4.4 beschriebenen Verfahren – im folgenden als „Mittelwert-Verfahren“ bezeichnet;
- c) das zweite Optimierungsproblem mit stochastischen Nebenbedingungen, szenarienbasiertem deterministischem Äquivalent und dem in Abschnitt 4.3.4.5 beschriebenen erweiterten Lagrange-Verfahren.

Alle drei Optimierungsprobleme sind in einem einzigen C++-Programm mit der Binärdatei

`secondOptimizationProblem.exe`

implementiert. Die Auswahl des Optimierungsproblems und zahlreiche Verfahrensparameter werden über eine *Parameterdatei* gesteuert. Sie wird im Abschnitt 4.4.1.1 beschrieben. Die Definition des Akquiseplans und der Akquise (und somit des Fallbeispiels), die den Optimierungsproblemen zugrunde liegen, erfolgt in vier *Definitionsdateien*. Diese Dateien werden im Abschnitt 4.4.1.2 beschrieben. Daran schließt sich im Abschnitt 4.4.1.3 die Beschreibung der von dem Programm erzeugten *Ausgabe-Dateien* an. Eine Erläuterung der *Konsolenausgabe* des Programms folgt im Abschnitt 4.4.1.4 und schließlich werden im Abschnitt 4.4.1.5 die *Klassenstruktur* des Computerprogramms sowie sein *Hauptprogramm* beschrieben²⁹⁾.

Zur besseren Unterscheidung stehen nachfolgend alle Namen von Dateien, von Parametern aus der Parameterdatei, von Klassen des Programms u. dgl. in der Schriftart **Courier**.

27. siehe <http://www.LHSystems.com>

28. <http://vieta.math.tu-cottbus.de/~klweber>

29. Die Binärdatei, die Parameterdatei, die Definitionsdateien des Akquiseplans sowie die Ausgabe-Dateien können grundsätzlich auch andere Bezeichnungen tragen als die hier angegebenen.

4.4.1.1 Parameterdatei

Nachfolgend wird ein Beispiel der Parameterdatei `parameter.txt` gezeigt. Die Erläuterungen sind nach jedem Parameter eingerückt und in der Schriftart **Times** eingefügt.

```
; =====
; Parameter Initialization File
; Standard name: 'parameter.txt'
; =====

; -----
; input/output parameters
; -----

Parameter zur Steuerung der Eingabe- und Ausgabedaten des Programms

FILE_NAME_BASE                = "<Path>\akquisePlan"
    Pfad und Namensstamm (das heißt Dateiname ohne Endung) der Definitionsdateien des
    Akquiseplans (absolut oder relativ zum Verzeichnis, in dem das Programm aufgerufen
    wird)
REDIRECT_CPLEX_OUTPUT         = "ON"
    Die von CPLEX erzeugte Ausgabe erscheint entweder in dem Kommando-Fenster30), in
    dem das Programm aufgerufen wird (OFF), oder sie wird in die Datei
    FILE_NAME_BASE.cpx umgeleitet (ON).
WRITE_MODEL_EACH_SCENARIO     = "OFF"
    Wenn das szenarienbasierte Mittelwert-Verfahren angewendet wird (siehe oben: Fall b);
    vgl. weiter unten: Parameter OPTIMIZATION_MODE), dann wird das Optimierungsmodell
    für jedes Szenario im LP-Format31) in die Datei FILE_NAME_BASE__<scenario
    no>.lp geschrieben (ON), oder es wird nicht in eine Datei geschrieben (OFF).
WRITE_SOLUTION_EACH_SCENARIO = "OFF"
    Wenn das szenarienbasierte Mittelwert-Verfahren angewendet wird (siehe oben: Fall b);
    vgl. weiter unten: Parameter OPTIMIZATION_MODE), dann wird für jedes Szenario die
    optimale Lösung in die Datei FILE_NAME_BASE__<scenario no>.lp geschrieben
    (ON), oder sie wird nicht in eine Datei geschrieben (OFF).
WRITE_MODEL_EACH_ITERATION    = "OFF"
    Wenn das erweiterte Lagrange-Verfahren angewendet wird (siehe oben: Fall c); vgl. weiter
    unten: Parameter OPTIMIZATION_MODE), dann wird nach jeder Verfahrenssiteration das
    Optimierungsmodell im LP-Format31) in die Datei FILE_NAME_BASE__<iteration
    no>.lp geschrieben (ON), oder es wird erst nach Abbruch der Iteration in die Datei
    FILE_NAME_BASE__<final iteration no>.lp geschrieben (OFF).
WRITE_SOLUTION_EACH_ITERATION = "OFF"
    Wenn das erweiterte Lagrange-Verfahren angewendet wird (siehe oben: Fall c); vgl. weiter
    unten: Parameter OPTIMIZATION_MODE), dann wird nach jeder Verfahrenssiteration das
    Optimierungsergebnis in die Datei FILE_NAME_BASE__<iteration no>.sol
    geschrieben (ON), oder es wird erst nach Abbruch der Iteration in die Datei
    FILE_NAME_BASE.sol geschrieben (OFF).
WRITE_TIMENODE_PREDS_SUCCS    = "OFF"
```

30. Gemeint ist der *Microsoft Windows* Befehlsinterpreter `cmd.exe`.

31. Das *LP*-Format ist ein spezifisches Beschreibungsformat von *ILOG CPLEX* für Optimierungsprobleme, das der üblichen mathematischen Beschreibungsweise ähnelt [78].

In jedem Programmdurchlauf wird eine Beschreibung des zeitlichen Akquiseplans (der den Optimierungsmodellen a) – c) zugrunde liegt in die Datei `FILE_NAME_BASE.tap` ausgegeben. Zu jedem Knoten des zeitlichen Akquiseplans werden entweder alle Vorgänger- und Nachfolger-Knoten angegeben (ON) oder nicht angegeben (OFF). Die Ausgabe von Vorgänger- und Nachfolger-Knoten ist z. B. zu Kontrollzwecken nützlich.

```
CORRECT_NAMES = "OFF"
```

Die Variablennamen, die bei Ausgabe von Optimierungsmodellen im LP-Format (siehe Erläuterung der Parameter: `WRITE_MODEL_EACH_SCENARIO`, `WRITE_MODEL_EACH_ITERATION`) geschrieben werden, enthalten Zeichen, die den Syntax-Regeln für LP-Dateiformate³¹⁾ widersprechen. Diese Zeichen werden durch Syntax-konforme Zeichen ersetzt (ON) oder bleiben unverändert (OFF). Die Variablen brauchen nur dann Syntax-konform zu sein, wenn das im LP-Format geschriebene Optimierungsproblem wieder von *CPLEX* eingelesen werden soll – z. B. zu Kontrollzwecken.

```
; -----
; base aquisition plan parameters
; -----
```

Parameter zur Definition eines Akquiseplans (Def. 3.1) und für die Simulation einer Akquise (vgl. Abschnitt 3.3)

```
AQUIRE_PLAN_NODE_NUMBER = "19"
```

Anzahl der Knoten des Akquiseplans; in Def. 3.1 mit N bezeichnet

```
AQUIRE_DURATION = "100"
```

```
AQUIRE_DURATION_ACCURACY = "0.0001"
```

Abbruchkriterien für die Berechnung der Übergangsmatrizen $P_{\omega}(t)$ (vgl. Satz 3.12) sowie der Präsenzwahrscheinlichkeiten $p(t)$ (Def. 3.6); die Berechnung wird abgebrochen, wenn entweder $t = \text{AQUIRE_DURATION}$ oder die Summe der Präsenzwahrscheinlichkeiten in den Endknoten größer oder gleich $1 - \text{AQUIRE_DURATION_ACCURACY}$ beträgt. Im letzten Fall ist t gleich dem Akquiseende gemäß Def. 3.11.

```
; -----
; time aquisition plan parameters
; -----
```

Parameter zur Definition eines zeitlichen Akquiseplans (Def. 4.26)

```
AQUIRE_INPUT_DURATION = "10"
```

Ende des Zuführungszeitraums der Akquise, t_{in} gemäß Def. 4.12

```
; -----
; optimization mode parameters
; -----
```

Parameter zur Auswahl des zu lösenden Optimierungsproblems

```
OPTIMIZATION_MODE = "MULTICOMMODITY"
```

Schalter zur Festlegung des zu lösenden Optimierungsproblems, siehe oben: Fall a): `MULTICOMMODITY`; Fall b): `SCENARIO_AVERAGE`; Fall c): `SCENARIO_LAGRANGE`

```
; -----
; general optimization problem parameters
; -----
```

grundlegende Parameter der zu lösenden Optimierungsprobleme

```

MAX_FLOW                                     = "10000"
  Obergrenze des Flusses durch den zeitlichen Akquiseplan (Def. 4.22); gleich der Kapazi-
  tätsbeschränkung des Quellknotens des zeitlichen Akquiseplans
MAX_NUMBER_OF_CUSTOMERS_IN_TIMENODE        = "10000"
  Parameter  $c \in \mathbb{N}$  der normierten Multinomialverteilung gemäß Def. 4.17; entspricht der
  maximalen Anzahl von Kunden, die sich gleichzeitig in einem Knoten des zeitlichen
  Akquiseplans befinden
SCENARIO_NUMBER                             = "1"
  Anzahl der Szenarien; gleich dem Stichprobenumfang  $N$  in (4.128) (siehe oben: Fall b) und
  in (4.140) (siehe oben: Fall c)
TRUTH_LEVEL                                 = "0.75"
  Wahrheitsniveau, für welches die Optimierungsprobleme b) und c) gelöst werden (vgl.
  Def. 4.21)

; -----
; numerical stability parameters
; -----
  Parameter zur Gewährleistung der numerischen Stabilität des Programms

EPS_ACCURACY                               = "1e-12"
  Genauigkeitsangabe für Rundungsoperationen, die im Programm zur Sicherung numeri-
  scher Stabilität durchgeführt werden.
CHECK_AQUISEPLAN_COMPUTEPROBS              = "OFF"
  Entweder werden die berechneten Präsenz- und Übergangswahrscheinlichkeiten (vgl.
  Abschnitt 3.3) auf ihre Genauigkeit überprüft (ON) oder nicht überprüft (OFF); dazu gehört
  z. B. die Überprüfung ob die Wahrscheinlichkeitswerte in der Zeile einer stochastischen
  Matrix gleich eins sind.
EPS_CALC_CHECK                             = "1e-12"
  Genauigkeitsangabe für die Überprüfung von Wahrscheinlichkeitswerten wenn der Parame-
  ter CHECK_AQUISEPLAN_COMPUTEPROBS den Wert ON besitzt.

; -----
; augmented Lagrangian method parameters
; -----
  Parameter des erweiterten Lagrange-Verfahrens (siehe oben Fall c); alle Angaben beziehen
  sich auf Alg. 4.6.

MAX_ITERATION                              = "20"
  Maximale Anzahl an Iterationen des erweiterten Lagrange-Verfahrens; wenn das Verfahren
  nicht vorher abbricht, wird es nach Erreichen von MAX_ITERATION Iterationen abgebro-
  chen.
LAMBDA_0                                   = "2"
  Anfangsparameter  $\lambda^{(0)} \geq 0$ 
LAMBDA_NON_NEGATIVE                        = "ON"
  Wenn die Verfahrensparameter  $\lambda^{(k+1)}$  negativ sind, dann wird ihr Wert auf null gesetzt
  (ON) oder bleibt, wie er ist (OFF).
LAMBDA_UPDATE_BETA                        = "ON"

```

Parameter zur Steuerung einer Verfahrensvariante:

Entweder gilt $\lambda^{(k+1)} := \lambda^{(k)} + C^{(k+1)} h^{(k+1)}$ (OFF) oder $\lambda^{(k+1)} := \lambda^{(k)} + \beta h^{(k+1)}$ (ON).

C_0 = "10"

Anfangsparameter $c^{(0)} \geq 0$

C_UPDATE_MAX = "OFF"

Parameter zur Steuerung einer Verfahrensvariante:

Entweder wird der komponentenweise Vergleich $|h_i^{(k+1)}| > \gamma |h_i^{(k)}|$ durchgeführt (OFF) oder es wird der Vergleich $|h_i^{(k+1)}| > \gamma \cdot \max_i |h_i^{(k)}|$ durchgeführt (ON).

C_UNIQUE_FLAG = "OFF"

Parameter zur Steuerung einer Verfahrensvariante:

Entweder wird die komponentenweise Berechnung $c_i^{(k+1)} := \beta c_i^{(k)}$ durchgeführt (OFF) oder alle Komponenten $c_i^{(k+1)}$ erhalten den gleichen Wert: $c_i^{(k+1)} := \beta \cdot \max_i c_i^{(k)}$ (ON).

C_MAX_FLAG = "ON"

Entweder können die Parameter $c_i^{(k+1)}$ beliebig groß werden (OFF) oder sie werden bei Überschreiten des Wertes C_MAX_VALUE auf diesen Wert gesetzt (ON).

C_MAX_VALUE = "100000"

Maximalwert der Parameter $c_i^{(k+1)}$, wenn C_MAX_FLAG den Wert ON besitzt.

BETA = "10"

Wert des Parameters $\beta > 1$

GAMMA = "0.25"

Wert des Parameters $\gamma > 0$

EPS = "0.001"

Wert des Parameters $\epsilon > 0$

EPS_MEAN_FLAG = "ON"

Parameter zur Steuerung einer Verfahrensvariante:

Entweder wird das Verfahren abgebrochen, wenn $\|h^{(k)}\| < \epsilon$ gilt (OFF) oder wenn der Mittelwert der $h_i^{(k)}$ kleiner als ϵ ist (ON).

ROUND_LINEAR_OBJECTIVE = "ON"

Entweder werden die Koeffizienten der linearen Bestandteile der Zielfunktion gerundet (ON) oder nicht (OFF); die Rundung wird durch den Parameter EPS_ACCURACY (siehe oben) gesteuert.

ROUND_SOLUTION = "ON"

Für die Optimierungsprobleme b) und c) werden die Lösungswerte der Optimierungsvariablen entweder gerundet (ON) oder nicht gerundet (OFF); die Rundung wird durch den Parameter EPS_ACCURACY (siehe oben) gesteuert.

; -----

; random number generation parameters

; -----

Parameter zur Steuerung der Zufallszahlengenerierung (vgl. Abschnitt 4.3.5)

RANDOM_REPRODUCE = "ON"

Die im Programmablauf erzeugten Zufallszahlen bilden entweder eine reproduzierbare Folge³²⁾ (ON), die allein vom Wert des Parameters `RANDOM_INIT_NUMBER` abhängt, oder bei jedem Programmaufruf wird eine andere Folge von Zufallszahlen erzeugt (OFF). Ohne reproduzierbare Zufallszahlen können die Ergebnisse verschiedener Optimierungsläufe nur bedingt miteinander verglichen werden.

`RANDOM_INIT_NUMBER` = "5000"

Wenn der Parameter `RANDOM_REPRODUCE` den Wert ON besitzt, dann legt diese Zahl die Folge der erzeugten Zufallszahlen eindeutig fest.

```
; -----  
; CPLEX parameters  
; -----
```

Die Bedeutung dieser *CPLEX*-Parameter (vgl. Fußnote 26 auf Seite 168) ist in *Appendix A Parameter Table* des Referenzhandbuchs von *CPLEX* beschrieben [78].

<code>CPX_PARAM_ADVIND</code>	= "1"	; default: 1
<code>CPX_PARAM_CLOCKTYPE</code>	= "1"	; default: 1
<code>CPX_PARAM_COLGROWTH</code>	= "100"	; default: 100
<code>CPX_PARAM_DEPIND</code>	= "1"	; default: 0
<code>CPX_PARAM_EPMRK</code>	= "0.99999"	; default: 0.01
<code>CPX_PARAM_LPMETHOD</code>	= "1"	; default: 0
<code>CPX_PARAM_NETFIND</code>	= "2"	; default: 2
<code>CPX_PARAM_QPNZREADLIM</code>	= "500"	; default: 500
<code>CPX_PARAM_QPNZGROWTH</code>	= "200"	; default: 200
<code>CPX_PARAM_ROWGROWTH</code>	= "100"	; default: 100
<code>CPX_PARAM_SCAIND</code>	= "0"	; default: 0
<code>CPX_PARAM_SCRIND</code>	= "1"	; default: 0
<code>CPX_PARAM_WORKDIR</code>	= "."	; default: '.'
<code>CPX_PARAM_WORKMEM</code>	= "128.0"	; default: 128.0

Neben der oben beschriebenen Parameterdatei sind vier Definitionsdateien nötig, um das zweite Optimierungsproblem genau zu beschreiben. Der Anschaulichkeit halber werden nachfolgend nicht die Definitionsdateien des Fallbeispiels mit dem Akquiseplan aus Abb. 3.5 gezeigt – ihre Einträge sind zu umfangreich –, sondern Definitionsdateien des Akquiseplans in Abb. 4.6.

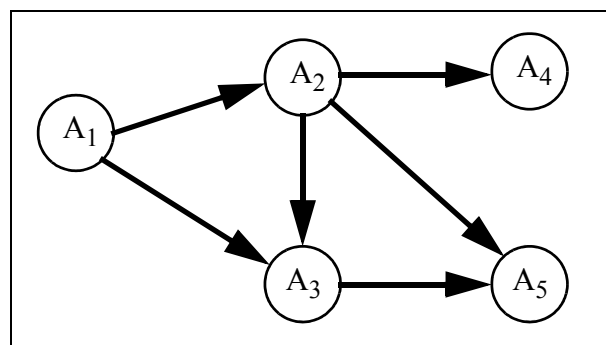


Abbildung 4.6: Beispiel eines Akquiseplans zur Erläuterung der Definitionsdateien.

32. Das Programm kann keine wirklichen Zufallszahlen erzeugen, sondern nur Pseudozufallszahlen.

4.4.1.2 Definitionsdateien

Definitionsdatei `aquisePlan.adj`

Ein Akquiseplan mit N Knoten wird durch seine Adjazenzmatrix $A = (a_{ij}) \in \{0, 1\}^{N \times N}$ festgelegt (vgl. Def. 3.1 und 3.2). Sie ist in der Datei `FILE_NAME_BASE.adj` enthalten. Für den Akquiseplan in Abb. 4.6 zeigt Abb. 4.7 den Inhalt der Datei. Der Wert für das Matrixelement a_{ij} steht in Zeile i und Spalte j .

0	1	1	0	0
0	0	1	1	1
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0

Abbildung 4.7: Definitionsdatei `aquisePlan.adj` für den Akquiseplan in Abb. 4.6.

Definitionsdateien `aquisePlan.trs` und `aquisePlan.tme`

Gemäß Satz 3.12 können die Vektoren der Präsenzwahrscheinlichkeiten $p(t)$ für einen gegebenen Anfangsvektor $p(0)$ iterativ mithilfe der zeitabhängigen Übergangsmatrizen $P_\omega(t+1)$ entsprechend der Vorschrift (3.25) berechnet werden. Die Übergangsmatrizen wiederum werden in jedem Iterationsschritt aus den

- Wahrscheinlichkeiten des Wechsels nach Ankunft im Knoten p_{ij}^t (Def. 3.8) und der
- zeitlichen Punktwahrscheinlichkeiten des Knotenwechsels $\varphi_{ij}(\tau)$ (Def. 3.9)

berechnet, wobei erstere als zeitinvariant angenommen wird. Die Werte für p_{ij} sind zu einer stochastischen Matrix zusammenfasst in der Definitionsdatei `aquisePlan.trs` enthalten. Für den Akquiseplan in Abb. 4.6 zeigt Abb. 4.8 den Inhalt der Datei. Die zeitlichen Punkt-

0	0.3	0.7	0	0
0	0	0.5	0.25	0.25
0	0	0	0	1
0	0	0	1	0
0	0	0	0	1

Abbildung 4.8: Definitionsdatei `aquisePlan.trs` für den Akquiseplan in Abb. 4.6.

wahrscheinlichkeiten sind für alle Kanten des Akquiseplans in der Definitionsdatei `aquisePlan.tme` enthalten. Für den Akquiseplan in Abb. 4.6 zeigt Abb. 4.9 den Inhalt der Datei. Die Syntax der Zeilen ist wie folgt:

- Eintrag 1: Knotenindex i ,
- Eintrag 2: Knotenindex j ,

1	2	1	2	1		
1	3	1	2	1		
2	3	2	1	0.5	2	0.5
2	4	1	2	1		
2	5	1	2	1		
3	5	2	1	0.5	2	0.5

Abbildung 4.9: Definitionsdatei `aquisePlan.tme` für den Akquiseplan in Abb. 4.6.

- Eintrag 3: Anzahl der Sprungstellen von $\varphi_{ij}(\tau)$ – nachfolgend mit n bezeichnet,
- Einträge 4 bis $4 + 2n - 1$: jeweils paarweise Angabe von $(\tau_k, \varphi_{ij}(\tau_k))$ für $k = 1, \dots, n$, wobei

$$\varphi_{ij}(\tau) = \begin{cases} 0 & \text{für } 0 \leq \tau < \tau_1 \\ \varphi_{ij}(\tau_k) & \text{für } \tau_k \leq \tau < \tau_{k+1} \\ 0 & \text{für } \tau_n < \tau \end{cases}.$$

Definitionsdatei `aquisePlan.fcp`

Für jede Akquiseaktion, das heißt für jeden Knoten des Akquiseplans mit Ausnahme der beiden Endknoten (vgl. Def. 3.1) gilt eine unscharfe Kapazitätsschranke (Def. 4.2), die in den Optimierungsproblemen für ein gegebenes Wahrheitsniveau zu berücksichtigen ist. Jede unscharfe Kapazitätsschranke ist durch eine unscharfe Menge μ_1 gegeben. Sie werden als parametrisierte trapezförmige unscharfe Mengen in der Definitionsdatei `aquisePlan.fcp` angegeben. Für den Akquiseplan in Abb. 4.6 zeigt Abb. 4.10 den Inhalt der Datei. Die Syntax

<code>trapIntv int</code>	0	0	100	200
<code>trapIntv int</code>	0	0	50	100
<code>trapIntv int</code>	0	0	200	300

Abbildung 4.10: Definitionsdatei `aquisePlan.fcp` für den Akquiseplan in Abb. 4.6.

der Zeilen ist wie folgt:

- Eintrag 1: Typ der unscharfen Menge; gegenwärtig kann das Programm nur unscharfe Mengen des Typs `trapIntv` („trapezoidal fuzzy interval“, trapezförmiges unscharfes Intervall) einlesen und verarbeiten.
- Eintrag 2: Typ des Trägers der unscharfen Menge (Def. 2.4); gegenwärtig kann das Programm nur unscharfe Mengen für Träger des Typs `int` („integer“, ganzzahlig) verarbeiten, das heißt $\text{suppl} \subseteq \mathbb{Z}$.
- Einträge 3 – 7: Stützstellen $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ der trapezförmigen unscharfen Menge μ_1 , so dass gilt:

$$\mu_1(x) = \begin{cases} 0 & \text{für } x < a_1 \\ \left\lfloor \frac{x - a_1}{a_2 - a_1} \right\rfloor & \text{für } (a_1 \neq a_2) \wedge (a_1 \leq x < a_2) \\ 1 & \text{für } (a_1 = a_2) \wedge (x = a_1) \\ 1 & \text{für } (a_2 \neq a_3) \wedge (a_2 \leq x < a_3) \\ 1 & \text{für } (a_2 = a_3) \wedge (x = a_2) \\ 1 - \left\lfloor \frac{a_3 - x}{a_3 - a_4} \right\rfloor & \text{für } (a_3 \neq a_4) \wedge (a_3 \leq x < a_4) \\ 0 & \text{für } (a_3 = a_4) \wedge (x = a_3) \\ 0 & \text{für } a_4 \leq x \end{cases}.$$

Wegen der spezifischen Eigenschaften von unscharfen Kapazitätsschranken (vgl. Def. 4.2) ist hier stets $a_1 = a_2 = 0$.

4.4.1.3 Ausgabe-Dateien

Das Programm erzeugt verschiedene Ausgabedateien, deren Steuerung von ihrem jeweiligen Zweck abhängt.

- a) Kopien der Definitionsdateien dienen allein Testzwecken und werden nur erzeugt, wenn die entsprechenden Methoden im Hauptprogramm (siehe Abschnitt 4.4.1.5) aufgerufen werden. Ihre Dateinamen sind Erweiterungen der Namen der eingelesenen Dateien:

- `aquisePlan.adj.cpy`,
- `aquisePlan.trs.cpy`,
- `aquisePlan.tme.cpy`,
- `aquisePlan.fcp.cpy`.

Der Inhalt dieser Dateien ist mit dem der Definitionsdateien identisch.

- b) Dateien, die Informationen über den Programmablauf und Ergebnisse enthalten und bei jedem Programmaufruf geschrieben werden:

- `aquisePlan.err`,
- `aquisePlan.prp`,
- `aquisePlan.tap`,

- c) Dateien, deren Ausgabe vom Optimierungsmodus abhängt, das heißt vom Wert des Parameters `OPTIMIZATION_MODE` (siehe Abschnitt 4.4.1.1) in der Parameterdatei.

- `aquisePlan.sol`;
wird nur für `OPTIMIZATION_MODE = "MULTICOMMODITY"` oder
`OPTIMIZATION_MODE = "SCENARIO_AVERAGE"` geschrieben;
- `aquisePlan.lp`;
wird nur für `OPTIMIZATION_MODE = "MULTICOMMODITY"` geschrieben;

- `aquisePlan___<final iteration no>.lp`;
wird nur für `OPTIMIZATION_MODE = "SCENARIO_LAGRANGE"` geschrieben;
- `aquisePlan.lgr`;
wird nur für `OPTIMIZATION_MODE = "SCENARIO_LAGRANGE"` geschrieben.

d) Dateien, die nur dann ausgegeben werden, wenn die zugehörigen Parameter in der Parameterdatei (siehe Abschnitt 4.4.1.1) gesetzt sind:

- `aquisePlan.cpx`;
nur für `REDIRECT_CPLEX_OUTPUT = "ON"`;
- `aquisePlan___<scenario no>.lp`;
nur für `WRITE_MODEL_EACH_SCENARIO = "ON"`;
- `aquisePlan___<iteration no>.lp`
nur für `WRITE_MODEL_EACH_ITERATION = "ON"`;
- `aquisePlan___<scenario no>.sol`;
nur für `WRITE_SOLUTION_EACH_SCENARIO = "ON"`;
- `aquisePlan___<iteration no>.sol`;
nur für `WRITE_SOLUTION_EACH_ITERATION = "ON"`.

Nachfolgend werden die Dateien aus b) – d) erläutert.

Fehlerdatei

Bei fehlerfreiem Programmverlauf enthält die Fehlerdatei `aquisePlan.err` die Angabe von Start- und Endzeit, z. B.

```
Sun Dec 05 15:32:42 2004 --- Application started.
Sun Dec 05 15:33:25 2004 --- Application terminated.
```

Bei fehlerfreiem Programmverlauf enthält sie entweder konkrete Hinweise auf die Fehlerursache oder stellt eine unbekannte Ausnahme (*unknown exception*) fest, z. B.

```
Sun Dec 05 16:53:30 2004 --- Application started.
Exception 'class FileExistExc' in FILE_NAME_BASE.cpp, line 415: <input file
does not exist>. [Sun Dec 05 16:53:31 2004]
...
Sun Dec 05 16:53:31 2004 --- Application terminated.
```

Datei der Präsenzwahrscheinlichkeiten

Die gemäß Satz 3.12 berechneten Vektoren der Präsenzwahrscheinlichkeiten $p(t)$ stehen spaltenweise in der Datei `aquisePlan.prp`. Sie werden für $t = 1, \dots, t_{\text{end}}$ berechnet (der Wert für $t = 0$ ist vorgegeben). Zur Berechnung von t_{end} siehe die Erläuterungen zu den Parametern `AQUISE_DURATION` und `AQUISE_DURATION_ACCURACY` in Abschnitt 4.4.1.1.

Datei mit dem zeitlichen Akquiseplan

Die Datei `aquisePlan.tap` enthält die Beschreibung des aus dem gegebenen Akquiseplan und verschiedenen vorgegebenen Parametern berechneten zeitlichen Akquiseplans (Def. 4.26).

Für den Akquiseplan in Abb. 4.6, die Definitionsdateien aus Abschnitt 4.4.1.2 sowie die Parameterwerte

```
AQUISE_DURATION = "20",
AQUISE_INPUT_DURATION = "2",
AQUISE_DURATION_ACCURACY = "0.0001"
```

zeigt Abb. 4.11 den Inhalt der Datei. Sie enthält im Abschnitt `time aquise nodes` eine

<pre>time aquise nodes ===== start time tau = -1 ----- (0, -1, -1) (6, -1, -1) start time tau = 0 ----- (1, 0, 0) (1, 1, 0) ... (5, 9, 0) start time tau = 1 ----- (1, 1, 1) (1, 2, 1) ... (5, 9, 1) start time tau = 2 ----- (1, 2, 2) (1, 3, 2) ... (5, 9, 2)</pre>	<pre>(Fortsetzung von links unten) time aquise edges ===== start time tau = -1 ----- (6, 0, -1, -1) start time tau = 0 ----- (0, 1, 0, 0) (4, 6, 2, 0) ... (5, 5, 8, 0) start time tau = 1 ----- (0, 1, 1, 1) (4, 6, 3, 1) ... (5, 5, 8, 1) start time tau = 2 ----- (0, 1, 2, 2) (4, 6, 4, 2) ... (5, 5, 8, 2)</pre>
(Fortsetzung rechts oben)	

Abbildung 4.11: Datei mit dem zeitlichen Akquiseplan, `aquisePlan.tap` für den Akquiseplan in Abb. 4.6 und bestimmte im Text angegebene Parametereinstellungen.

Beschreibung der Knoten und im Abschnitt `time aquise edges` eine Beschreibung der Kanten des zeitlichen Akquiseplans. Darin sind jeweils für alle Zuführungszeitpunkte $\tau = 0, 1, \dots, t_{\text{in}}$ (hier: $t_{\text{in}} = 2$) und für die vom Programm berechnete Akquisedauer (hier $t_{\text{end}} = 9$)

- die Knoten in Indexschreibweise (vgl. Anmerkung 5 zu Def. 4.26) und
- die Kanten in Indexschreibweise (vgl. Anmerkung 13 zu Def. 4.26)

aufgeführt. Die Zuführungszeitpunkt $\tau = -1$ gibt es nur der Form halber wegen der besonderen Indizierung der Quelle A_0 und der Senke A_{N+1} des zeitlichen Akquiseplans (vgl. Anmerkung 2 zu Def. 4.26). Unter dem Zuführungszeitpunkt $\tau = -1$ finden sich im Abschnitt `time aquise nodes` diese beiden Knoten und im Abschnitt `time aquise edges` die Kante (A_{N-1}, A_0) .

Wenn der Parameter `WRITE_TIMENODE_PREDS_SUCCS` den Wert `ON` besitzt, dann werden zusätzlich zu jedem Knoten des zeitlichen Akquiseplans alle Vorgänger- und Nachfolgerknoten ausgegeben. Den Auszug einer solchermaßen erweiterten Datei `aquisePlan.tap` zeigt Abb. 4.12. Darin stehen für jeden Knoten unter seinem Index

```
time aquise nodes
=====

...

start time tau = 0
-----

(1, 0, 0)
preds: (0, -1, -1)
succs: (1, 1, 0) (2, 1, 0) (3, 1, 0)
...
(2, 1, 0)
preds: (1, 0, 0)
succs: (2, 2, 0) (3, 2, 0) (4, 2, 0) (5, 2, 0)
(2, 2, 0)
preds: (1, 1, 0) (2, 1, 0)
succs: (2, 3, 0) (3, 3, 0) (4, 3, 0) (5, 3, 0)
(2, 3, 0)
preds: (1, 2, 0) (2, 2, 0)
succs: (2, 4, 0) (3, 4, 0) (4, 4, 0) (5, 4, 0)
...
```

Abbildung 4.12: Datei mit dem zeitlichen Akquiseplan `aquisePlan.tap` für den Akquiseplan in Abb. 4.6 und die Parametereinstellung `WRITE_TIMENODE_PREDS_SUCCS = "ON"`.

- hinter dem Schlüsselwort `preds` („predecessors“) seine Vorgängerknoten,
- hinter dem Schlüsselwort `succs` („successors“) seine Nachfolgerknoten (vgl. Def. 4.27).

Modelldateien

Die Modelldateien mit der Endung `*.lp` enthalten eine Beschreibung des an *CPLEX* übergebenen Optimierungsproblems im *ILOG*-spezifischen *LP-Format* (siehe Abschnitt 4.4.1.1, Fußnote 31). Je nach Wert der eingangs erwähnten Parameter wird eine Modelldatei

- einmalig zum Programmende geschrieben oder
- für jedes Szenario bzw. jede Iteration.

Ergebnisdateien

Der Inhalt der Ergebnisdateien mit der Endung *.sol hängt vom Wert des Parameters OPTIMIZATION_MODE ab.

- Für die Werte "MULTICOMMODITY" oder "SCENARIO_LAGRANGE" enthalten Ergebnisdateien den Wert der Zielfunktion sowie für jede Optimierungsvariable ihren Optimalwert. Ein Beispiel für den Akquiseplan in Abb. 4.6 zeigt Abb. 4.13. Unter den Optimierungsvariablen sind nur jene wesentlich, die einem Kantenfluss entsprechen, der im Quellknoten des zeitlichen Akquiseplans entspringt. Diese Kantenflüsse sind die Lösung des zweiten Optimierungsproblems. Die korrespondierenden Optimierungsvariablen werden nachfolgend als „wesentliche Optimierungsvariable“ bezeichnet.

```
objective value = -3875

(6, 0, -1, -1)          = 125
(0, 1, 0, 0)            = 52.9328
(0, 1, 1, 1)            = 40.9701
(0, 1, 2, 2)            = 31.097
(4, 6, 2, 0)            = 1.80503e-009
(5, 6, 2, 0)            = 2.37323e-009
(4, 6, 3, 0)            = 1.51534e-009
(5, 6, 3, 0)            = 2.72288e-009
(4, 6, 4, 0)            = 2.64916e-009
(5, 6, 4, 0)            = 4.80475e-009
(4, 6, 5, 0)            = 4.41903e-009
(5, 6, 5, 0)            = 5.2221
(4, 6, 6, 0)            = 0.868815
(5, 6, 6, 0)            = 8.44659
(4, 6, 7, 0)            = 1.17352
(5, 6, 7, 0)            = 10.9613
(4, 6, 8, 0)            = 0
(5, 6, 8, 0)            = 0
...
```

Abbildung 4.13: Beispiel einer Ergebnisdatei *.sol (Auszug) für den Akquiseplan in Abb. 4.6 und OPTIMIZATION_MODE = "SCENARIO_LAGRANGE"; zu jeder Kante (Indizes links) steht rechts der Optimalwert ihres Kantenflusses (dem eine Optimierungsvariable entspricht). Die wesentlichen Optimierungsvariablen sind gelb unterlegt.

- Für die Werte "SCENARIO_AVERAGE" kann es bei entsprechender Parameterwahl für jedes Szenario eine Ergebnisdatei geben, deren Struktur mit jener für die Werte "MULTICOMMODITY" oder "SCENARIO_LAGRANGE" übereinstimmt. Diese Dateien enthalten im Dateinamen die Szenarionummer. Daneben gibt es unabhängig von der Parameterwahl die Ergebnisdatei aquisePlan.sol, die für jedes Szenario die Optimalwerte der wesentlichen Optimierungsvariablen (siehe oben) sowie den daraus gebildeten Durchschnitt (siehe Abschnitt 4.3.4.4) enthalten. Ein Beispiel für den Akquiseplan in Abb. 3.5, 10 Szena-

rien und die Zuführungszeitpunkte $\tau = 0, 1, \dots, 5$ zeigt Abb. 4.14. Darin entspricht jedem Zuführungszeitpunkt τ eine wesentliche Optimierungsvariable.

tau	:	0	1	2	3	4	5
Scenario No 1:		63.08	55.87	55.87	55.87	55.87	145.70
Scenario No 2:		73.60	52.91	52.91	52.91	52.91	124.61
Scenario No 3:		61.35	61.35	61.35	58.31	51.40	128.65
Scenario No 4:		74.68	74.68	53.76	53.76	53.76	115.54
Scenario No 5:		79.49	52.63	52.63	52.63	52.63	137.82
...							
Scenario No 16:		79.71	51.81	51.81	51.81	51.81	142.97
Scenario No 17:		83.96	56.97	53.76	53.76	53.76	109.01
Scenario No 18:		56.50	56.50	56.50	56.50	56.30	129.61
Scenario No 19:		59.95	59.17	59.17	59.17	59.17	131.33
Scenario No 20:		59.52	59.52	59.52	59.52	57.45	128.52
Average	:	68.00	56.77	55.57	55.42	54.67	129.85
(rounded up)	:	69.00	57.00	56.00	56.00	55.00	130.00

Abbildung 4.14: Ergebnisdatei `aquisePlan.sol` für den Akquiseplan in Abb. 3.5, 20 Szenarien und die Zuführungszeitpunkte $\tau = 0, 1, \dots, 5$. Jedem Zuführungszeitpunkt entspricht eine wesentliche Optimierungsvariable.

Lagrange-Protokolldatei

Die Lagrange-Protokolldatei `aquisePlan.lgr` dokumentiert für jede Iteration $k = 1, 2, \dots$ des Lagrange-Verfahrens

- die Werte der wesentlichen Optimierungsvariablen,
- die Werte der Verfahrensparameter $\lambda^{(k)}$ und $c^{(k)}$ des Alg. 4.6,
- den Zielfunktionswert,
- die Residuen der Kunden-Verteilungs-Nebenbedingungen.

CPLEX-Protokolldatei

Die *CPLEX*-Protokolldatei `aquisePlan.cpx` enthält Angaben zu den internen Lösungsschritten von *CPLEX*, die von der Art des Problems abhängen. Sie unterscheiden sich z. B. für lineare Optimierungsprobleme und für quadratische Optimierungsprobleme. Diese Angaben sind teilweise selbst erklärend. In den *CPLEX*-Handbüchern finden sich keine näheren Erklärungen dazu.

4.4.1.4 Konsolenausgabe

Die Konsolenausgabe³³⁾ ist selbsterklärend. Sie kommentiert jeden Programmschritt, alle wesentlichen Kenngrößen des Problems sowie zu jedem Programmschritt die seit Programm-

start verstrichene Laufzeit mit Millisekunden-Genauigkeit. Sie enthält auch eine Übersicht für die Laufzeit des *CPLEX*-Algorithmus innerhalb des Anwendungsprogramms. Im Abschnitt 4.4.2.1 zeigt Abb. 4.16 das Beispiel einer Konsolenausgabe.

4.4.1.5 Beschreibung des Computerprogramms

Das Computerprogramm ist im *C++*-Quellcode kommentiert. Daneben gibt es eine *HTML*-Dokumentation, die mit der Dokumentationssoftware *Doxygen*³⁴⁾ erstellt worden ist und auf der Homepage des Verfassers abgelegt ist³⁵⁾. Der Quellcode ist in so genannten Klassen organisiert. Die Klassen sind im folgenden mit ihren wichtigsten Eigenschaften beschrieben. Das Verständnis der Grundbegriffe objektorientierter Softwareentwicklung wird vorausgesetzt³⁶⁾. Die *HTML*-Dokumentation ist umfassender und enthält

- eine Auflistung aller Dateien,
- eine Aufzählung aller Klassen und Strukturen,
- eine textuelle und grafische Darstellung der Klassenhierarchie,
- eine alphabetische Auflistung aller Klassenelemente,
- eine alphabetische Auflistung aller Dateielemente.

Zum besseren Verständnis des Quellcodes ist zu beachten, dass das Programm für das Optimierungsmodell nebeneinander sowohl eigene Datenstrukturen aufbaut als auch Datenstrukturen der *CPLEX C++*-Schnittstelle nutzt, und dass es hierbei teilweise Überschneidungen gibt. So besteht neben dem Attribut `OPTPROBLEM2::optVars` des Typs `vector<OPTVARTE>`, das die Optimierungsvariablen enthält auch das Attribut `OPTPROBLEM2::pOrigVariables` vom Typ `IloNumVarArray`, der eine Klasse der *CPLEX C++*-Schnittstelle ist.

AQUIREPLAN

- Die Klasse repräsentiert einen Akquiseplan (Def. 3.1).
- Mit ihren Methoden `readAdjaceMatrix`, `readTransProbMatrix` und `readTimeTransMatrix` werden die Definitionsdateien eines Akquiseplans (siehe Abschnitt 4.4.1.2) eingelesen. Die Methode `computeProbs` berechnet die Präsenzwahrscheinlichkeiten entsprechend Satz 3.12.

TIMEAQUIREPLAN

- Die Klasse ist von der Klasse `AQUIREPLAN` abgeleitet.

33. Gemeint ist die Ausgabe des *Microsoft Windows* Befehlsinterpreters `cmd.exe`.

34. *Doxygen* ist ein Dokumentationssystem für verschiedene Programmiersprachen, unter anderem *C++*, das aus den Quelldateien des Programms eine Dokumentation in verschiedenen Formaten erzeugen kann, darunter *HTML* [152].

35. <http://vieta.math.tu-cottbus.de/~klweber>

36. Eine Erklärung gibt jedes in *C++* oder eine andere objektorientierte Programmiersprache einführende Buch, z. B. [100].

- Sie repräsentiert einen Akquiseplan sowie entsprechend dem Wert des Parameters `AQUIRE_INPUT_DURATION` (siehe Abschnitt 4.4.1.1) den darauf aufbauenden zeitlichen Akquiseplan (Def. 4.26).

CAPACITYCON

- Die Klasse repräsentiert eine Kapazitätsschranke, die scharf oder unscharf sein kann. Als unscharfe Kapazitätsschranke ist sie über eine unscharfe Menge (Attribut `pFuzzySet` vom Typ `FUZZYSET*`) definiert und stellt für entsprechende unscharfe Mengen eine unscharfe Kapazitätsschranke gemäß Def. 4.2 dar.
- Im Fall einer unscharfen Kapazitätsschranken gibt ihr `operator()` zu dem Wahrheitswert `truthLevel` das Bild der verallgemeinerten Umkehrabbildung (Def. 4.3) für `pFuzzySet` zurück.

FUZZYSET

- Die Klasse repräsentiert eine unscharfe Menge (Def. 2.2).
- Die vorhandene Implementierung lässt nur trapezförmige parametrisierte unscharfe Mengen über Trägern aus \mathbb{Z} zu. Sie wird von der Klasse `CAPACITYCON` verwendet.

INTDISTRIB

- Die Klasse repräsentiert eine Wahrscheinlichkeitsverteilung über einer Teilmenge von \mathbb{Z} .
- Sie wird von der Klasse `AQUIREPLAN` verwendet, um die Vektoren der Präsenzwahrscheinlichkeit $p(t)$ (Def. 3.6), die Übergangsmatrizen $P_{\omega}(t)$ (Satz 3.12) und andere Wahrscheinlichkeitsverteilungen darzustellen.

OPTPROBLEM2

- Die Klasse repräsentiert das zweite Optimierungsproblem (Abschnitt 4.3.4).
- Sie enthält einen zeitlichen Akquiseplan (Attribut `optTimeAquirePlan` vom Typ `TIMEAQUIREPLAN*`), die Optimierungsvariablen (Attribut `optVars` vom Typ `vector<OPTVARTE>`³⁷⁾), drei Arten von Nebenbedingungen: Fluss-Erhaltungs-Nebenbedingungen `flowConsCons` (Def. 4.24, Anmerkung 4.), Knoten-Kapazitäts-Nebenbedingungen `nodeCapaCons` (gleich den Bündel-Nebenbedingungen, siehe Abschnitt 4.3.4.3) sowie die Kunden-Verteilungs-Nebenbedingungen `flowDiviCons` (siehe Abschnitte 4.3.4.4 und 4.3.4.5), und sie enthält Zeiger auf alle Klassen der *Cplex C++*-Schnittstelle³⁸⁾ zur Darstellung des Optimierungsproblems.
- Sie besitzt Methoden zum Einlesen der unscharfen Kapazitätsschranken (`readFuzzyCapCons`) aus der entsprechenden Definitionsdatei des Akquiseplans (siehe Abschnitt 4.4.1.2) und zur Erzeugung der oben aufgeführten Nebenbedingungen (`setFlowConsCons`, `setNodeCapaCons`, `setFlowDiviCons`) sowie zur Erzeugung der linearen Zielfunktion (`setOrigObj`).

37. `OPTVARTE` ist gleich der Klasse `OPTVAR<TIMEEDGE>`, siehe die Beschreibung von `OPTVAR<T>`.

38. siehe Fußnote 26 auf Seite 168

OPTSOLVER2

- Die Klasse ist von der Klasse `OPTPROBLEM2` abgeleitet.
- Mit ihren Methoden `setObjective` und `setConstraints` legt sie in Abhängigkeit vom Wert des Parameters `OPTIMIZATION_MODE` (siehe Abschnitt 4.4.1.1) für das jeweilige Optimierungsproblem und die ggf. spezifizieren Szenarien (siehe Abschnitt 4.4.2.2) oder Iterationen (siehe Abschnitt 4.4.2.3) Zielfunktion und Nebenbedingungen mit den Klassen der *CPLEX C++*-Schnittstelle³⁹⁾ an. Die Methode `startOptimization` steuert je nach Optimierungsproblem die aufeinander folgende Lösung der Szenarien-spezifischen linearen Optimierungsprobleme (`OPTIMIZATION_MODE = „SCENARIO_AVERAGE“`) bzw. die Iterationen des Lagrange-Verfahrens (`OPTIMIZATION_MODE = „SCENARIO_LAGRANGE“`) oder löst einmalig das Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen (`OPTIMIZATION_MODE = „MULTICOMMODITY“`).
- Sie enthält ein Objekt der Klasse `IloCplex` der *CPLEX C++*-Schnittstelle³⁹⁾ mit deren Methode `solve` die *CPLEX*-Algorithmen zur Lösung des Optimierungsproblems aufgerufen werden. Der Aufruf von `solve` wird von der Methode `startOptimization` (siehe oben) gesteuert.

OPTVAR<T>

- Diese Templateklasse repräsentiert eine Optimierungsvariable, die mit einem Objekt der Klasse `T` verbunden ist. In der vorliegenden Anwendung ist `T` gleich der Klasse `TIMEEDGE`, da jeder Optimierungsvariable ein Kantenfluss des zeitlichen Akquiseplans entspricht.
- Zu den Attributen der Klasse gehören neben dem Zeiger auf das Objekt der Klasse `T` auch das Attribut `value`, das in vorliegender Anwendung gleich dem Wert des Kantenflusses ist.

RANDOM

- Die Klasse stellt (Pseudo-) Zufallszahlen bereit (siehe Abschnitt 4.3.5), die je nach Wert des Parameters (siehe Abschnitt 4.4.1.1) `RANDOM_REPRODUCE` reproduzierbar oder einmalig sind.
- Die Methoden `getUniform`, `getBinomial` und `getMultinomial` geben gleichverteilte, binomialverteilte oder multinomialverteilte Zufallszahlen zurück.

SCENARIO

- Die Klasse ist von der Klasse `RANDOM` abgeleitet.
- Die Klasse repräsentiert in dem Attribut `sample` ein Szenario der Kundenverteilung für einen Akquiseplan `acquirePlan` und die über den Parameterwert `AQUIRE_DURATION` spezifizierte Akquisedauer.

39. siehe Fußnote 26 auf Seite 168

- Die Methode `getSample` gibt für jede Kante des Akquiseplans und jeden Zeitpunkt den Wert von $\xi_{ij}(t)$ zurück (vgl. Def. 4.18).

TIMEEDGE

- Die Klasse repräsentiert eine Kante eines zeitlichen Akquiseplans (Def. 4.26).
- Sie enthält unter anderem die Attribute `start` und `end` vom Typ `TIMENODE*`, die die die Kante begrenzenden Knoten des zeitlichen Akquiseplans repräsentieren.

TIMENODE

- Die Klasse repräsentiert einen Knoten eines zeitlichen Akquiseplans (Def. 4.26).
- Sie enthält unter anderem die Attribute `predecessors` und `successors` vom Typ `vector<TIMENODE*>`, die die Vorgängerknoten und Nachfolgerknoten des Knotens repräsentieren.

Hauptprogramm `main.cpp`

Das Hauptprogramm ist in Abb. 4.15 dargestellt. Es besteht in der ersten Hälfte aus der Deklaration von Variablen und dem Einlesen von Parameterwerten (`cParameter::GetValue`). Danach wird ein zeitlicher Akquiseplan erzeugt, das heißt ein Objekt der Klasse `TIMEAQUISEPLAN` angelegt (`tAP`). Das Ergebnis, der zeitliche Akquiseplan und die Präsenzwahrscheinlichkeiten werden in Dateien protokolliert (`tAP->writeTimeAquirePlan()`, `tAP->writePresenceProbs()`). Danach wird das Optimierungsproblem im Hauptspeicher abgebildet und gelöst, das heißt es wird ein Objekt der Klasse `OPTSOLVER2` (und damit auch der Basisklasse `OPTPROBLEM2`) angelegt (`optSolve2`). Das Lösungsverfahren wird vom Methodenaufruf `optSolve2.startOptimization()` angestoßen.

4.4.2 Untersuchungsprogramm

Die Fallbeispiele in den nachfolgenden Abschnitten dienen der Illustration der Lösungsverfahren für das zweite Optimierungsproblem in der Darstellung aus Abschnitt 4.3.4, der Untersuchung ihrer Eigenschaften sowie dem Vergleich und der Bewertung der Verfahren. Diese Aspekte werden systematisch untersucht. Eine Übersicht über das Untersuchungsprogramm gibt Tab. 4.2. Sie enthält neben der Beschreibung der Untersuchungen die Vorsilben, die eine Zuordnung der vom Computerprogramm erzeugten Dateien und manuell erstellten *Excel*⁴⁰⁾-Auswertungstabellen ermöglicht. Aus Platzgründen werden in den folgenden Abschnitten nur die Ergebnisse dargestellt. Auf die Dateien aller Untersuchungen kann über die Homepage des Verfassers⁴¹⁾ zugegriffen werden. Der Einfachheit halber werden nachfolgend die Lösungsverfahren für das Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen (das sind die Nebenbedingungen der stochastischen Verteilung der Kunden in den Knoten) wie folgt bezeichnet:

- „Mittelwert-Verfahren“ (beschrieben im Abschnitt 4.3.4.4),

40. *Excel* ist ein Tabellenkalkulationsprogramm der Fa. *Microsoft* (<http://www.microsoft.de>). Verwendet wurde die Version *Excel 2000*.

41. <http://vieta.math.tu-cottbus.de/~klweber>


```

...
int main(int argc, char* argv[]) {

    int            aquireDuration;
    int            aquireInputDuration;
    int            aquirePlanNodeNumber;
    int            maxFlow;
    double         truthLevel;
    string         help;
    string         parameterFileName;
    string         aquireFileName;

    ...
    // read parameter file
    cParameter::Initialize(parameterFileName);

    ...
    // read parameter values
    cParameter::GetValue("FILE_NAME_BASE", aquireFileName);
    cParameter::GetValue("AQUIRE_DURATION", help);
    CALL( aquireDuration = atoi(help.c_str()) );
    cParameter::GetValue("AQUIRE_INPUT_DURATION", help);
    CALL( aquireInputDuration = atoi(help.c_str()) );
    cParameter::GetValue("AQUIRE_PLAN_NODE_NUMBER", help);
    CALL( aquirePlanNodeNumber = atoi(help.c_str()) );
    cParameter::GetValue("TRUTH_LEVEL", help);
    CALL( truthLevel = atof(help.c_str()) );
    cParameter::GetValue("MAX_FLOW", help);
    CALL( maxFlow = atoi(help.c_str()) );

    TIMEAQUIREPLAN* tAP;
    tAP = new TIMEAQUIREPLAN(
        aquirePlanNodeNumber,
        aquireDuration,
        aquireFileName,
        aquireInputDuration);
    CALL( tAP->writeTimeAquirePlan() );
    CALL( tAP->writePresenceProbs() );

    OPTSOLVER2 optSolve2(
        tAP,
        truthLevel,
        maxFlow);
    CALL( optSolve2.startOptimization() );

    ...
} // int main(int argc, char* argv[])

```

Abbildung 4.15: Hauptprogramm main.cpp.

- „Lagrange-Verfahren“ (beschrieben im Abschnitt 4.3.4.5).

Die Fallbeispiele wurden mit einem Personalcomputer (Desktop) und einem Laptop mit folgender Hardware- und Betriebssystemkonfiguration gerechnet:

- Prozessor (CPU)
 - Desktop: AMD Athlon XP 1700+ mit 1,47 GHz

Fall	Optimierungsproblem	Verfahren ^{a)}	Abschnitt	Vorsilbe ^{b)}
1.	Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen, einmalige Zuführung von Kunden	LP	4.4.2.1	MCF1<y>_
2.	wie Fall 1., aber mit mehrmaliger Zuführung von Kunden	LP	4.4.2.1	MCF2<y>_
3.	Vergleich der Fälle 1. und 2.		4.4.2.1	
4.	Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen, einmalige Zuführung von Kunden	MV	4.4.2.2	AVG1<y>_
5.	wie Fall 4., aber mit dem Lagrange-Verfahren	LV	4.4.2.3	LGR1<y>_
6.	Vergleich der Fälle 4. und 5.		4.4.2.3	
7.	wie Fall 4., aber mit mehrmaliger Zuführung von Kunden	MV	4.4.2.2	AVG2<y>_
8.	wie Fall 5., aber mit mehrmaliger Zuführung von Kunden	LV	4.4.2.3	LGR2<y>_
9.	Vergleich der Fälle 7. und 8.		4.4.2.3	
10.	Laufzeitverhalten für das Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen	MV	4.4.2.2	AVGR<y>_
11.	wie Fall 10., aber mit dem Lagrange-Verfahren	LV	4.4.2.3	LGRR<y>_
12.	Vergleich der Fälle 10. und 11.		4.4.2.3	
13.	Abhängigkeit der Lösung des Mehrgüterfluss-Problems mit stochastischen Nebenbedingungen vom Wahrheitsniveau	MV	4.4.2.2	AVGT<y>_
14.	wie Fall 13., aber mit dem Lagrange-Verfahren	LV	4.4.2.3	LGRT<y>_
15.	Vergleich der Fälle 13. und 14.		4.4.2.3	xzyT<y>_

Tabelle 4.2: Untersuchungsprogramm für die Verfahren zur Lösung des zweiten Optimierungsproblems.

a. LP: Simplex-Algorithmus, MV: Mittelwert-Verfahren, LV: Lagrange-Verfahren

b. Systematik: Buchstaben/Ziffern 1 – 3: Verfahren: MCF: Mehrgüterfluss-Problem (*Multi-Commodity Flow Problem*), AVG: Mittelwertbildung (*Averaging*), LGR: erweitertes Lagrange-Verfahren (*Augmented Lagrange Method*); Buchstabe/Ziffer 4: Untersuchungsparameter: 1: einmalige Zuführung von Kunden, 2: mehrmalige Zuführung von Kunden, R: Laufzeitverhalten (*Runtime Behaviour*), T: Wahrheitswert (*Truth Value*); Buchstabe/Ziffer 5 (optional): Mehrere Untersuchungen des gleichen Falles werden durch Anhängen der Buchstaben a – z unterschieden.

- Laptop: Intel Pentium 4, Mobile CPU mit 1,80 GHz
- Hauptspeicher (RAM)
 - Desktop: 1 GB
 - Laptop: 768 MB

- Festplattenspeicher
 - Desktop: 2 x 55,8 GB, gespiegelt
 - Laptop: 27,9 GB
- Betriebssystem
 - Desktop: Windows XP Professional, Version 2002, SP 2
 - Laptop: Windows XP Home Edition, Version 2002, SP 2

Während der Programmausführung waren keine weiteren Anwendungsprogramme aktiv. Alle Laufzeituntersuchungen wurden auf demselben Rechner und unter den gleichen Auslastungsbedingungen durchgeführt. Versuche haben gezeigt, dass die Leistungsfähigkeit beider Rechner bezüglich der Laufzeit von Programmen ungefähr gleich ist, so dass alle Ergebnisse unabhängig vom Rechner, mit dem sie ermittelt wurden, verglichen werden können.

Der besseren Übersichtlichkeit halber sind die Fälle in drei Unterabschnitten zusammengefasst:

- Lösung des zweiten Optimierungsproblems als Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen im Abschnitt 4.4.2.1,
- Lösung des zweiten Optimierungsproblems als Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen mit dem Mittelwert-Verfahren im Abschnitt 4.4.2.2,
- Lösung des zweiten Optimierungsproblems als Mehrgüterfluss-Problem mit stochastischen Nebenbedingungen mit dem Lagrange-Verfahren im Abschnitt 4.4.2.3.

Für die Nebenbedingungen werden folgende Bezeichnungen verwendet:

- „Fluss-Erhaltungs-Nebenbedingungen“⁴²⁾,
- „Knoten-Kapazitäts-Nebenbedingungen“ für die Bündel-Nebenbedingungen⁴³⁾,
- „Kunden-Verteilungs-Nebenbedingungen“ für die Nebenbedingungen der stochastischen Verteilung der Kunden in den Knoten⁴⁴⁾.

Die Konsolenausgabe (vgl. Abschnitt 4.4.1.4) wird im Programmaufruf in die Textdatei mit der Bezeichnung `<Vorsilbe>console.txt` umgeleitet. Wo erforderlich werden *Excel*⁴⁵⁾-Auswertungsdateien mit der Bezeichnung `<Vorsilbe>analysis.xls` erstellt.

Die Fälle werden anhand wesentlicher Merkmale tabellarisch beschrieben, darunter

- das Laufzeitverhalten⁴⁶⁾ und

42. siehe Def. 4.24, Anmerkung 4.; in der Konsolenausgabe: *flow conservation constraints*

43. siehe Abschnitt 4.3.4.3; in der Konsolenausgabe: *node capacity constraints*

44. siehe Abschnitte 4.3.4.4 und 4.3.4.5; in der Konsolenausgabe: *flow division constraints*

45. siehe Fußnote 40 auf Seite 186

46. Die Begriffe „Laufzeitverhalten“, „Laufzeit“ („runtime“) und „Rechenzeit“ werden in gleicher Bedeutung verwendet.

- die Optimierungsergebnisse.

4.4.2.1 Lösung ohne stochastische Nebenbedingungen

Das zweite Optimierungsproblem als Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen berücksichtigt weder das zeitliche Verhalten der Kunden noch die Verschiedenartigkeit ihrer Reaktionen auf eine Akquiseaktion. Kunden in einem Knoten werden so auf die Folgeknoten verteilt, dass der Fluss im Akquiseplan (vgl. Def. 4.22) maximal wird. Dieser Fluss unterliegt nur den Fluss-Erhaltungs-Nebenbedingungen (vgl. Abschnitt 4.3.4.2) und den Bündel-Nebenbedingungen, die die unscharfen Kapazitätsschranken widerspiegeln (vgl. Abschnitt 4.3.4.3). Der praktische Nutzen der Lösung dieses Problems liegt darin, dass sie eine obere Schranke der Lösung mit stochastischen Nebenbedingungen darstellt.

Fall 1

Dieser Sonderfall des zweiten Optimierungsproblems ohne stochastische Nebenbedingungen entspricht dem ersten Optimierungsproblem ohne stochastische Nebenbedingungen (vgl. Abschnitt 3.4).

Datei-Vorsilbe	MCF1_
OPTIMIZATION_MODE	"MULTICOMMODITY"
AQUISE_DURATION	"100"
AQUISE_INPUT_DURATION	"0"
SCENARIO_NUMBER	"1"
TRUTH_LEVEL	"0.75"
Akquisedauer	75
zeitlicher Akquiseplan: Anzahl der Knoten	1377
zeitlicher Akquiseplan: Anzahl der Kanten	4812
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	1377
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	1230
Anzahl der Kunden-Verteilungs-Nebenbedingungen	0
Laufzeit (gesamt)	1,482s
Laufzeit (CPLEX-Optimierung)	0,12s 8,1 % der Gesamtlaufzeit
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	t = 0: 562
Ergebnis / Anmerkungen	
Laufzeitverhalten: Die Rechenzeit ist vernachlässigbar. Sie wird hauptsächlich für das Einlesen der Definitionsdateien, das Schreiben von Ergebnis- und Protokolldateien sowie den Aufbau der Datenstrukturen, darunter die des zeitlichen Akquiseplans verwendet (siehe Konsolenausgabe des Programms in Abb. 4.16).	

Optimierungsergebnis: Da die Schlupfvariable $s(1, 0)$ der Knoten-Kapazitäts-Nebenbedingung gleich null ist, wird der maximale Fluss des zeitlichen Akquiseplans durch die Kapazitätsbeschränkung der Akquiseaktion 1 begrenzt (vgl. `MCF1_aquisePlan.sol`).

Dies folgt auch aus der unscharfen Kapazitätsschranke \tilde{l} von Akquiseaktion 1:

$$\mu_1^{-1}(0,75) = 562,5 \text{ (MCF1_aquisePlan.fcp).}$$

Anmerkung: Die Anzahl der Fluss-Erhaltungs-Nebenbedingungen stimmt mit der Anzahl der Knoten des zeitlichen Akquiseplans überein (`MCF1_console.txt`). Da in jedem Knoten die Divergenz gleich null sein muss (vgl. Def. 4.22) erzeugt das Programm für jeden Knoten eine Fluss-Erhaltungs-Nebenbedingung.

			(Fortsetzung von links unten)		
***	AQUIREPLAN: instantiate object	***			
***		0.0000 sec	***		
***	AQUIREPLAN: read adjacency matrix	***			
***		0.0000 sec	***		
***	AQUIREPLAN: adjacency matrix read	***	***	OPTPROBLEM2: read fuzzy capacity constraints	***
***		0.0000 sec	***		0.8410 sec
***	AQUIREPLAN: read transition probabilities	***	***	OPTPROBLEM2: fuzzy capacity constraints read	***
***		0.0000 sec	***		0.8410 sec
***	AQUIREPLAN: transition probabilities read	***	***	OPTPROBLEM2: set original objective function	***
***		0.0000 sec	***		0.8410 sec
***	AQUIREPLAN: read time distributions of node	***	***	OPTPROBLEM2: original objective function set	***
***	transitions	***	***		0.8410 sec
***		0.0000 sec	***	OPTPROBLEM2: set flow conservation constraints	***
***	AQUIREPLAN: time distributions of node transitions	***	***		0.8410 sec
***	read	***	***	OPTPROBLEM2: 1377	***
***		0.0100 sec	***	flow conservation constraints set	***
***	AQUIREPLAN: compute probabilities of presence	***	***		0.9010 sec
***		0.0100 sec	***	OPTPROBLEM2: set node capacity constraints	***
***	AQUIREPLAN: probabilities of presence computed	***	***		0.9010 sec
***	for t = 0, ..., 75	***	***	OPTPROBLEM2: 1230	***
***		0.4700 sec	***	node capacity constraints set	***
***	AQUIREPLAN: object instantiated	***	***		0.9510 sec
***		0.4800 sec	***	OPTPROBLEM2: object instantiated	***
***	TIMEAQUIREPLAN: instantiate object	***	***		0.9510 sec
***		0.4800 sec	***	OPTSOLVER2: instantiate object	***
***	TIMEAQUIREPLAN: aquire duration = 75	***	***	OPTSOLVER2: instantiate CPLEX model and algorithm	***
***		0.4800 sec	***		1.0010 sec
***	TIMEAQUIREPLAN: create time nodes	***	***	OPTSOLVER2: set CPLEX objective	***
***		0.4800 sec	***		1.0910 sec
***	TIMEAQUIREPLAN: 1377 time nodes created	***	***	OPTSOLVER2: CPLEX objective set	***
***		0.4800 sec	***		1.1010 sec
***	TIMEAQUIREPLAN: create time edges	***	***	OPTSOLVER2: set CPLEX constraints	***
***		0.4800 sec	***		1.1010 sec
***	TIMEAQUIREPLAN: 4812 time edges created	***	***	OPTSOLVER2: CPLEX constraints set	***
***		0.5700 sec	***		1.1210 sec
***	TIMEAQUIREPLAN: object instantiated	***	***	OPTSOLVER2: CPLEX model and algorithm instantiated	***
***		0.5700 sec	***		1.1210 sec
***	TIMEAQUIREPLAN: write time aquisition plan	***	***	OPTSOLVER2: object instantiated	***
***		0.5700 sec	***		1.1210 sec
***	TIMEAQUIREPLAN: time aquisition plan written	***	***	OPTPROBLEM2: write copy of fuzzy capacity	***
***		0.6400 sec	***	constraints	***
***	AQUIREPLAN: write copy of adjacency matrix	***	***		1.1210 sec
***		0.6400 sec	***	OPTPROBLEM2: copy of fuzzy capacity constraints	***
***	AQUIREPLAN: copy of adjacency matrix written	***	***	written	***
***		0.6400 sec	***		1.1310 sec
***	AQUIREPLAN: write copy of transition probabilities	***	***	OPTSOLVER2: optimization started	***
***	matrix	***	***		1.1410 sec
***		0.6400 sec	***	OPTSOLVER2: optimization finished; solution status	***
***	AQUIREPLAN: copy of transition probabilities matrix	***	***	is: Optimal	***
***	written	***	***		1.2610 sec
***		0.6500 sec	***	OPTSOLVER2: write optimization model	***
***	AQUIREPLAN: write copy of time distributions of	***	***		1.2610 sec
***	node transitions	***	***	OPTSOLVER2: optimization model written	***
***		0.6500 sec	***		1.3010 sec
***	AQUIREPLAN: copy of time distributions of node	***	***	OPTSOLVER2: write optimization solution	***
***	transitions written	***	***		1.3010 sec
***		0.6500 sec	***	OPTSOLVER2: optimization solution written	***
***	AQUIREPLAN: write vectors of probabilities of	***	***		1.4620 sec
***	presence for whole aquisition time	***	***	OPTSOLVER2: CPLEX runtime information:	***
***		0.6500 sec	***	scenario/iteration no	***
***	AQUIREPLAN: vectors of probabilities of presence	***	***	1	0.12 sec
***	written for whole aquisition time	***	***		***
***		0.6600 sec	***	overall runtime:	0.12 sec
***	OPTPROBLEM2: instantiate object	***	***	overall runtime:	8.10 %
***		0.6600 sec	***	minimal runtime:	0.12 sec
***	OPTPROBLEM2: optimization mode is MULTICOMMODITY	***	***	maximal runtime:	0.12 sec
***		0.6600 sec	***	mean runtime:	0.12 sec
***	OPTPROBLEM2: set optimization variables	***	***	std dev runtime:	0.00 sec

192

Fall 2

Datei-Vorsilbe	MCF2_
OPTIMIZATION_MODE	"MULTICOMMODITY"
AQUIRE_DURATION	"100"
AQUIRE_INPUT_DURATION	"5"
SCENARIO_NUMBER	"1"
TRUTH_LEVEL	"0.75"
Akquisedauer	80
zeitlicher Akquiseplan: Anzahl der Knoten	8537
zeitlicher Akquiseplan: Anzahl der Kanten	29872
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	8537
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	1315
Anzahl der Kunden-Verteilungs-Nebenbedingungen	0
Laufzeit (gesamt)	8,311s
Laufzeit (CPLEX-Optimierung)	1,1s 13,3 % der Gesamtlaufzeit
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	t = 0: 562 t = 1, ..., 5: jeweils 313 Summe: 2127
Ergebnis / Anmerkungen	
Laufzeitverhalten: Die Rechenzeit der CPLEX-Optimierung ist sehr kurz. Es gilt das in Fall 1 Gesagte. Mit 3,155s nimmt der Aufbau der Kanten des zeitlichen Akquiseplans den größten Anteil an der Gesamtlaufzeit ein (vgl. MFC2_console.txt).	
Optimierungsergebnis: Der maximale Fluss des zeitlichen Akquiseplans wird durch die Kapazitätsbeschränkung der Akquiseaktion 1 begrenzt. Dies trifft auch für die Zuführungszeitpunkte $t = 1, \dots, 5$ zu, da die Schlupfvariablen der entsprechenden Fluss-Erhaltungs-Nebenbedingungen alle gleich null sind (vgl. MCF2_aquisePlan.sol).	

Fall 3

Im Fall 2 ist der Anteil der CPLEX-Optimierung an der Gesamtlaufzeit etwas größer als im Fall 1. Der Vergleich der Optimierungsergebnisse zeigt, dass der Fluss im zeitlichen Akquiseplan einer „Stauwirkung“ unterliegt, da zu den Zeitpunkten $t = 1, \dots, 5$ weniger Kunden zugeführt werden als die Kapazitätsschranke der ersten Aktion erlaubt. Das wird deutlich an zahlreichen Schlupfvariablen mit dem Wert null (siehe MCF2_aquisePlan.sol).

4.4.2.2 Lösung mit stochastischen Nebenbedingungen

Fall 4

Datei-Vorsilbe	AVG1_
OPTIMIZATION_MODE	"SCENARIO_AVERAGE"
AQUIRE_DURATION	"100"
AQUIRE_INPUT_DURATION	"0"
SCENARIO_NUMBER	"20"
TRUTH_LEVEL	"0.75"
Akquisedauer	75
zeitlicher Akquiseplan: Anzahl der Knoten	1377
zeitlicher Akquiseplan: Anzahl der Kanten	4812
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	1377
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	1230
Anzahl der Kunden-Verteilungs-Nebenbedingungen	4514 für jedes Szenario
Laufzeit (gesamt)	1min 53s
Laufzeit (CPLEX-Optimierung)	Laufzeit-Durchschnitt für ein Szenario ^{a)} , Laufzeit für alle Szenarien (absoluter Wert), Laufzeit für alle Szenarien (Anteil an der Gesamtlaufzeit ^{b)}): AVG1_: 0,86s 17,22s 15,18 %
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	aufgerundeter Wert für t = 0: 187
Ergebnis / Anmerkungen	
Laufzeitverhalten: (vgl. AVG1_console.txt) Wie in den Fällen 1 und 2 ist die Rechenzeit der CPLEX-Optimierung für ein Szenario sehr kurz. Den größten Teil der Gesamtlaufzeit macht mit 53 % die Erzeugung der 20 Szenarien aus (vgl. AVG1_console.txt).	
Optimierungsergebnis: Im Gegensatz zum Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen (Fall 1) sind die Lösungswerte nicht ganzzahlig und wesentlich kleiner als in jenem Fall (vgl. AVG1_aquisePlan.sol). Es gibt Schlupfvariablen von Knoten-Kapazitäts-Nebenbedingungen, die gleich null sind. Daran wird deutlich, dass auch die gegenüber Fall 1 geringere Anzahl zugeführter Kunden durch „Flaschenhälse“ im zeitlichen Akquiseplan bestimmt wird (vgl. AVG1_aquisePlan__1.sol – AVG1_aquisePlan_20.sol).	

Anmerkungen: Für jedes Szenario wurden jeweils eine Modell- und eine Ergebnisdatei ausgegeben (vgl. Abschnitt 4.4.1.3). Dadurch verlängert sich die Gesamtlaufzeit

Die Kunden-Verteilungs-Nebenbedingungen unterscheiden sich für jedes Szenario. Die anderen Nebenbedingungen stimmen für alle 20 Szenarien überein.

Ein Vergleich der *CPLEX*-Protokolldatei `AVG1_aquisePlan.cpx` mit den entsprechenden Dateien der Fälle 1 und 2 zeigt, dass das Optimierungsproblem dieses Falles offensichtlich komplexer ist, da *CPLEX* für jedes Szenario zahlreiche Iterationen benötigt und zur Beschleunigung der Lösung den Störungsmechanismus („perturbation“) einsetzt.

Der große Laufzeitanteil für die Erzeugung der Szenarien zeigt, dass das dreistufige Verfahren zur Erzeugung multinomialverteilter Zufallsvektoren (vgl. Abschnitt 4.3.5) verbesserungswürdig ist.

- a. Der Durchschnittswert wird über alle Optimierungsläufe berechnet. Die Werte sind der Datei `AVGRx_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Szenarien und die Standardabweichung der Laufzeit.
- b. Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (vgl. Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

Fall 7

Datei-Vorsilbe	AVG2_
OPTIMIZATION_MODE	"SCENARIO_AVERAGE"
AQUISE_DURATION	"100"
AQUISE_INPUT_DURATION	"5"
SCENARIO_NUMBER	"20"
TRUTH_LEVEL	"0.75"
Akquisedauer	80
zeitlicher Akquiseplan: Anzahl der Knoten	7967
zeitlicher Akquiseplan: Anzahl der Kanten	27862
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	7967
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	1230
Anzahl der Kunden-Verteilungs-Nebenbedingungen	26139 für jedes Szenario
Laufzeit (gesamt)	5min 20s
Laufzeit (CPLEX-Optimierung)	Laufzeit-Durchschnitt für ein Szenario ^{a)} , Laufzeit für alle Szenarien (absoluter Wert), Laufzeit für alle Szenarien (Anteil an der Gesamtlaufzeit ^{b)}): AVG2_: 7,69s 153,88s 48,10 %
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	für t = 0, 1, ..., 5 jeweils aufgerundet sowie die Summe der aufgerundeten Werte: 69, 57, 56, 56, 55, 130; Summe: 423
Ergebnis / Anmerkungen	
Laufzeitverhalten: Die Rechenzeit der CPLEX-Optimierung macht über die Hälfte der Gesamtlaufzeit aus. Die Erzeugung aller Szenarien beträgt etwa 28 % der Gesamtlaufzeit (vgl. AVG2_console.txt).	
Optimierungsergebnis: (vgl. AVG2_console.sol) Im Gegensatz zum Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen (Fall 2) sind die Lösungswerte nicht ganzzahlig und wesentlich kleiner als in jenem Fall. Es gibt Schlupfvariablen von Knoten-Kapazitäts-Nebenbedingungen, die gleich null sind. Daran wird deutlich, dass auch die gegenüber Fall 2 geringere Anzahl zugeführter Kunden durch „Flaschenhälse“ im zeitlichen Akquiseplan bestimmt wird (vgl. AVG1_aquisePlan__1.sol – AVG1_aquisePlan_20.sol).	

Anmerkungen: Die Kunden-Verteilungs-Nebenbedingungen unterscheiden sich für jedes Szenario. Die anderen Nebenbedingungen stimmen für alle 20 Szenarien überein.

Ein Vergleich der *CPLEX*-Protokolldatei `AVG2_aquisePlan.cpx` mit den entsprechenden Dateien der Fälle 1, 2 und 4 zeigt, dass das Optimierungsproblem dieses Falles offensichtlich komplexer ist, da *CPLEX* für jedes Szenario noch mehr Iterationen benötigt und zur Beschleunigung der Lösung den Störungsmechanismus („perturbation“) einsetzt.

Der große Laufzeitanteil für die Erzeugung der Szenarien zeigt, dass das dreistufige Verfahren zur Erzeugung multinomialverteilter Zufallsvektoren (vgl. Abschnitt 4.3.5) verbesserungswürdig ist.

- Der Durchschnittswert wird über alle Optimierungsläufe berechnet. Die Werte sind der Datei `AVGRx_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Szenarien und die Standardabweichung der Laufzeit.
- Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (siehe Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

Fall 10

Untersucht wird die Abhängigkeit der Laufzeit von der Anzahl der Szenarien und von der Dauer der Zuführung von Kunden (gegeben durch die Werte der Parameter `AQUISE_INPUT_DURATION`, `SCENARIO_NUMBER`). Eine Übersicht der betrachteten Kombinationen beider Parameter bzw. der mit diesen Parameterkombinationen durchgeführten Optimierungsläufe zeigt Abb. 4.17. Zur Ermittlung der Abhängigkeit der Rechenzeit von der

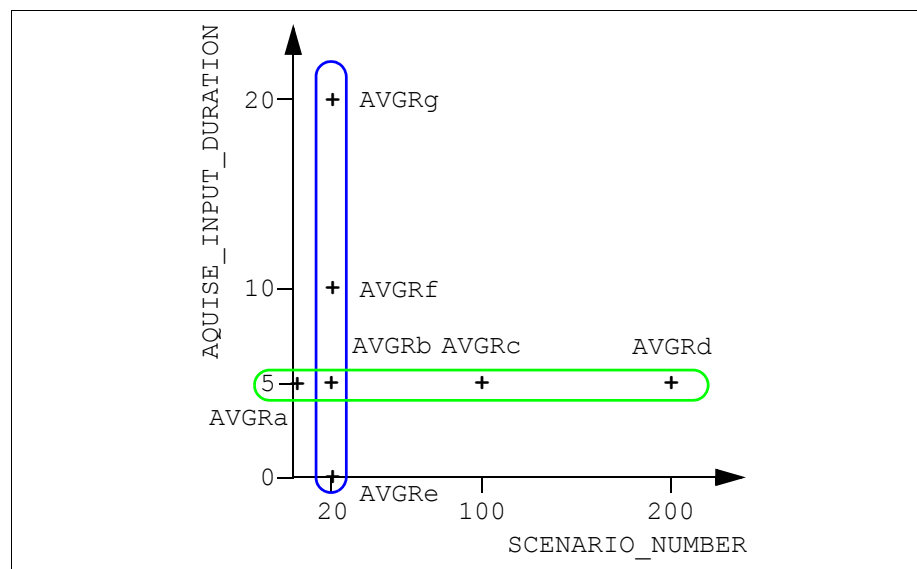


Abbildung 4.17: Fall 10: Untersuchte Kombinationen aus Szenarienzahl (Parameter `SCENARIO_NUMBER`) und Dauer der Zuführung von Kunden (Parameter `AQUISE_INPUT_DURATION`), der Vergleich wird zwischen den Fällen im blauen Rahmen und zwischen den Fällen im grünen Rahmen durchgeführt.

Anzahl der Szenarien werden die Kombinationen `AVGRa_` – `AVGRd_` untereinander verglichen, zur Ermittlung der Abhängigkeit der Rechenzeit von der Dauer der Kundenzuführung

die Kombinationen AVGRb_ und AVGRe_ – AVGRg_ Die Einzelergebnisse sind in folgender Tabelle zusammengestellt.

Datei-Vorsilbe	AVGRa_ – AVGRg_
OPTIMIZATION_MODE	"SCENARIO_AVERAGE"
AQUISE_DURATION	AVGRa_ – AVGRg_ : "100"
AQUISE_INPUT_DURATION	AVGRa_ – AVGRd_ : "5" AVGRe_ : "0" AVGRf_ : "10" AVGRg_ : "20"
SCENARIO_NUMBER	AVGRa_ : "1" AVGRb_ : "20" AVGRc_ : "100" AVGRd_ : "200" AVGRe_ – AVGRg_ : "20"
TRUTH_LEVEL	"0.75"
Akquisedauer	AVGRa_ – AVGRd_ : 80 AVGRe_ : 75 AVGRf_ : 85 AVGRg_ : 95
zeitlicher Akquiseplan: Anzahl der Knoten	AVGRa_ – AVGRd_ : 8537 AVGRe_ : 1377 AVGRf_ : 16172 AVGRg_ : 32867
zeitlicher Akquiseplan: Anzahl der Kanten	AVGRa_ – AVGRd_ : 29872 AVGRe_ : 4812 AVGRf_ : 56607 AVGRg_ : 115102
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	AVGRa_ – AVGRd_ : 8537 AVGRe_ : 1377 AVGRf_ : 16172 AVGRg_ : 32867
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	AVGRa_ – AVGRd_ : 1315 AVGRe_ : 1230 AVGRf_ : 1400 AVGRg_ : 1570

Anzahl der Kunden-Verteilungs-Nebenbedingungen	Die Werte gelten jeweils für ein Szenario. AVGRa_ – AVGRd_ : 27537 AVGRe_ : 4514 AVGRf_ : 50602 AVGRg_ : 96732																												
Laufzeit (gesamt)	AVGRa_ : 25s AVGRb_ : 4min, 38s AVGRc_ : 22min, 25s AVGRd_ : 50min, 49s AVGRe_ : 1min, 46s AVGRf_ : 11min, 32s AVGRg_ : 48min, 23s																												
Laufzeit (CPLEX-Optimierung)	jeweils Laufzeit-Durchschnitt für ein Szenario ^{a)} , Laufzeit für alle Szenarien (absoluter Wert), Laufzeit für alle Szenarien (Anteil an der Gesamtlaufzeit ^{b)}): <table><tr><td>AVGRa_ :</td><td>5,91s</td><td>5,91s</td><td>23,45 %</td></tr><tr><td>AVGRb_ :</td><td>7,71s</td><td>154,18s</td><td>55,34 %</td></tr><tr><td>AVGRc_ :</td><td>7,53s</td><td>752,56s</td><td>55,97 %</td></tr><tr><td>AVGRd_ :</td><td>7,62s</td><td>1523,95s</td><td>54,85 %</td></tr><tr><td>AVGRe_ :</td><td>0,82s</td><td>16,33s</td><td>15,4 %</td></tr><tr><td>AVGRf_ :</td><td>25,85s</td><td>517,03s</td><td>74,68 %</td></tr><tr><td>AVGRg_ :</td><td>128,69s</td><td>2573,74s</td><td>88,65 %</td></tr></table>	AVGRa_ :	5,91s	5,91s	23,45 %	AVGRb_ :	7,71s	154,18s	55,34 %	AVGRc_ :	7,53s	752,56s	55,97 %	AVGRd_ :	7,62s	1523,95s	54,85 %	AVGRe_ :	0,82s	16,33s	15,4 %	AVGRf_ :	25,85s	517,03s	74,68 %	AVGRg_ :	128,69s	2573,74s	88,65 %
AVGRa_ :	5,91s	5,91s	23,45 %																										
AVGRb_ :	7,71s	154,18s	55,34 %																										
AVGRc_ :	7,53s	752,56s	55,97 %																										
AVGRd_ :	7,62s	1523,95s	54,85 %																										
AVGRe_ :	0,82s	16,33s	15,4 %																										
AVGRf_ :	25,85s	517,03s	74,68 %																										
AVGRg_ :	128,69s	2573,74s	88,65 %																										

Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	für $t = 0, 1, \dots$ jeweils aufgerundet sowie die Summe der aufgerundeten Werte: AVGRa_ (1 Szenario) 64, 56, 56, 56, 56, 146; Summe: 434 AVGRb_ (20 Szenarien) 69, 57, 56, 56, 55, 130; Summe: 423 AVGRc_ (100 Szenarien) 69, 58, 56, 55, 54, 133; Summe: 425 AVGRd_ (200 Szenarien) 68, 58, 57, 55, 54, 133; Summe: 425 AVGRE_ (20 Szenarien) 187; Summe: 187 AVGRf_ (20 Szenarien) 145, 61, 47, 49, 52, 33, 30, 44, 43, 51, 128; Summe: 683 AVGRg_ (20 Szenarien) 81, 76, 70, 59, 56, 71, 20, 30, 45, 56, 79, 49, 22, 21, 23, 70, 69, 45, 42, 44, 131; Summe: 1159
Ergebnis / Anmerkungen	
<p>Laufzeitverhalten: Bei gleicher Dauer der Kundenzuführung ist die durchschnittliche <i>CPLEX</i>-Laufzeit unabhängig von der Anzahl der Szenarien (AVGRa_ – AVGRd_). Die etwas geringere Laufzeit für AVGRa_ liegt in der Schwankungsbreite der Fälle AVGRb_ – AVGRd_.</p> <p>Für gleiche Dauer der Kundenzuführung wachsen die Gesamtlaufzeit und die <i>CPLEX</i>-Laufzeit annähernd linear mit der Anzahl der Szenarien (siehe Abb. 4.18). Daher ist der Anteil der <i>CPLEX</i>-Laufzeit an der Gesamtlaufzeit annähernd konstant. Dass dieser Anteil im Fall AVGRa_ unterdurchschnittlich ist kann an den Besonderheiten des einen Szenarios in diesem Fall liegen.</p> <p>Bei konstanter Szenarienzahl wächst die durchschnittliche <i>CPLEX</i>-Laufzeit (und damit auch die <i>CPLEX</i>-Laufzeit insgesamt) nichtlinear an. Verschiedene Quotienten aus der Anzahl von Knoten und Kanten des zeitlichen Akquiseplans bzw. Anzahlen von Nebenbedingungen mit der <i>CPLEX</i>-Laufzeit sind in Tab. 4.3 zusammengestellt. Sie zeigt, dass nur Quotienten mit Produkten oder Zweierpotenzen im Nenner für unterschiedliche Zuführungsdauern ungefähr dieselbe Größenordnung besitzen. Daraus kann gefolgert werden, dass die Laufzeit mit dem <i>Produkt</i> aus der Anzahl der Knoten und der Anzahl der Kanten des zeitlichen Akquiseplans (Quotient 12) bzw. dem <i>Quadrat</i> der Anzahl der Kanten (entsprechen den Optimierungsvariablen) (Quotient 13) bzw. dem <i>Quadrat</i> der Summe der Anzahl der Nebenbedingungen (Quotient 14) bzw. der Summe aus dem <i>Quadrat</i> der Anzahl der Kanten (Optimierungsvariablen) und dem <i>Quadrat</i> der Anzahl der Nebenbedingungen (Quotient 15) wächst.</p>	

Optimierungsergebnis: Bei gleicher Dauer der Kundenzuführung (in den betrachteten Fällen $AVGRa_ - AVGRd_$ beträgt sie fünf) ändert sich das Optimierungsergebnis ab 20 Szenarien (in den Fällen $AVGRb_ - AVGRd_$) nur unwesentlich. Für nur ein Szenario ($AVGRa_$) weicht es jedoch deutlich von diesem Ergebnis ab.

Für die Fälle $AVGRb_ , AVGRE_ - AVGRg_$ ist der Zusammenhang zwischen der Dauer der Kundenzuführung und der optimalen Anzahl zuzuführender Kunden in Abb. 4.20 grafisch dargestellt. Eine Regressionsanalyse ergibt folgende Regressionsgerade:

Kundenanzahl = $48,8 \cdot \text{Zuführungsdauer} + 137,4$. Das bedeutet, dass für gegebenen Kapazitätsschranken der Akquiseaktionen die Gesamtzahl der bewerbbaeren Kunden bei einmaliger Kundenzuführung bis auf einen Wert von ungefähr 186 gesteigert werden kann (vgl. auch Fall 4). Bei mehrmaliger Kundenzuführung können mit jeder zusätzlichen Zuführung jeweils etwa 49 Kunden mehr bearbeitet werden.

Anmerkungen: Zeitliche Aqusepläne mit derselben Dauer der Kundenzuführung ($AVGRa_ - AVGRd_$) besitzen dieselbe Anzahl Knoten und Kanten. Dementsprechend stimmen auch die Anzahlen der Fluss-Erhaltungs-Nebenbedingungen, der Knoten-Kapazitäts-Nebenbedingungen und der Kunden-Verteilungs-Nebenbedingungen für diese Optimierungsprobleme überein.

Mit zunehmender Dauer der Kundenzuführung ($AVGRa_ , AVGRE_ - AVGRg_$) wachsen auch die Akquisedauer, die Anzahl der zeitlichen Knoten und der zeitlichen Kanten, sowie die Anzahlen der Nebenbedingungen.

- Der Durchschnittswert wird über alle Optimierungsläufen berechnet. Die Werte sind der Datei `AVGRx_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Szenarien und die Standardabweichung der Laufzeit.
- Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (siehe Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

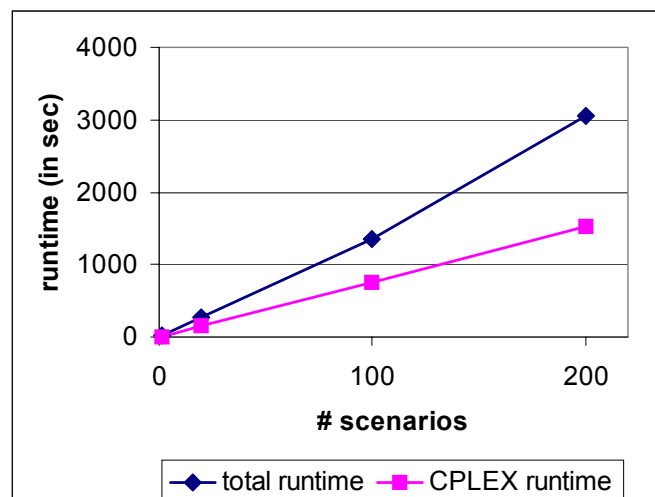


Abbildung 4.18: Grafische Darstellung der Gesamtlaufzeit des Programms und der Laufzeit der *CPLEX*-Optimierung in Abhängigkeit von der Anzahl der Szenarien für $AVGRa_ - AVGRd_$.

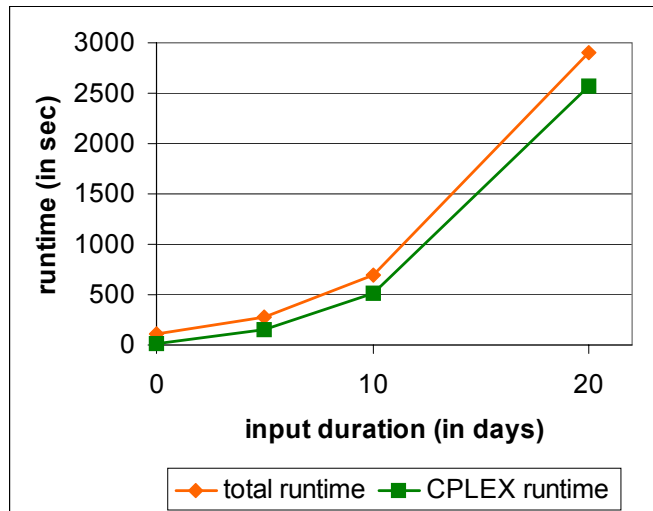


Abbildung 4.19: Grafische Darstellung der Gesamtlauzeit des Programms und der Laufzeit der *CPLEX*-Optimierung in Abhängigkeit von der Dauer der Kundenzuführung für $AVGRb_$, $AVGRe_ - AVGRg_$.

lfd. Nr.	Größe/ Quotient ^{a)}	Dauer der Kundenzuführung ^{b)}			
		0	5	10	20
1.	<i>CPLEX</i> -Laufzeit in s	16	154	517	2574
2.	Anzahl Knoten	1377	8537	16172	32867
3.	Anzahl Kanten	4812	29872	56607	115102
4.	Anzahl der Fluss-Erhaltungs-Nebenbedingungen	1377	8537	16172	32867
5.	Anzahl der Knoten-Kapazitäts-Nebenbedingungen	1230	1315	1400	1570

Tabelle 4.3: Quotienten aus Größen des zweiten Optimierungsproblems bzw. des zugehörigen zeitlichen Akquiseplans und der *CPLEX*-Laufzeit; die grün unterlegten Quotienten besitzen für positive Dauern der Kundenzuführung die gleiche Größenordnung.

lfd. Nr.	Größe/ Quotient ^{a)}	Dauer der Kundenzuführung ^{b)}			
		0	5	10	20
6.	Anzahl der Kunden-Verteilungs-Nebenbedingungen	4514	27537	50602	96732
7.	Quotient 2. / 1.	86,1	55,4	31,3	12,8
8.	Quotient 3. / 1.	300,8	194,0	109,5	44,7
9.	Quotient 4. / 1.	86,1	55,4	31,3	12,8
10.	Quotient 5. / 1.	76,9	8,5	2,7	0,6
11.	Quotient 6. / 1.	282,1	178,8	97,9	37,6
12.	Quotient (2. * 3.) / 1.	414,1	1656,0	1770,7	1469,7
13.	Quotient $3.^2 / (1. * 1000)$	1447,2	5794,4	6198,0	5147,0
14.	Quotient $(4. + 5. + 6.)^2 / (1. * 1000)$	3169,3	9077,5	8989,7	6684,3
15.	Quotient $[3.^2 + (4. + 5. + 6.)^2] / (1. * 1000)$	4616,5	14871,9	15187,7	11831,3

Tabelle 4.3: Quotienten aus Größen des zweiten Optimierungsproblems bzw. des zugehörigen zeitlichen Akquisepans und der *CPLEX*-Laufzeit; die grün unterlegten Quotienten besitzen für positive Dauern der Kundenzuführung die gleiche Größenordnung.

a. Ordnungszahlen in den Quotienten beziehen sich auf die angegebenen laufenden Nummern.

b. Wert des Parameters `AQUIRE_INPUT_DURATION`

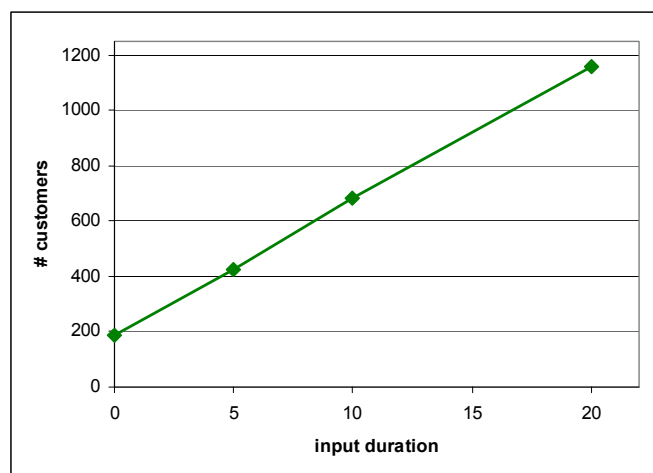


Abbildung 4.20: Grafische Darstellung der Anzahl der zugeführten Kunden (also des Kundendurchsatzes unter Beachtung der Kapazitäts-Nebenbedingungen) in Abhängigkeit von der Dauer der Kundenzuführung bei einer konstanten Anzahl von 20 Szenarien (`AVGRb_`, `AVGRg_`).

Fall 13

Datei-Vorsilbe	AVGTa_ – AVGTj_
OPTIMIZATION_MODE	"SCENARIO_AVERAGE"
AQUISE_DURATION	AVGTa_ – AVGTj_ : "100"
AQUISE_INPUT_DURATION	AVGTa_ – AVGTj_ : "10"
SCENARIO_NUMBER	AVGTa_ – AVGTj_ : "100"
TRUTH_LEVEL	AVGTa_ : "0.1" AVGTb_ : "0.2" AVGTC_ : "0.3" AVGTD_ : "0.4" AVGTE_ : "0.5" AVGTF_ : "0.6" AVGTG_ : "0.7" AVGTH_ : "0.8" AVGTI_ : "0.9" AVGTJ_ : "1.0"
Akquisedauer	AVGTa_ – AVGTj_ : 85
zeitlicher Akquiseplan: Anzahl der Knoten	AVGTa_ – AVGTj_ : 14082
zeitlicher Akquiseplan: Anzahl der Kanten	AVGTa_ – AVGTj_ : 49237
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	AVGTa_ – AVGTj_ : 14082
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	AVGTa_ – AVGTj_ : 1230
Anzahl der Kunden-Verteilungs-Nebenbedingungen	Der Wert gilt jeweils für ein Szenario. AVGTa_ – AVGTj_ : 46189
Laufzeit (gesamt)	AVGTa_ : 58min 38s AVGTb_ : 59min 18s AVGTC_ : 58min 3s AVGTD_ : 57min 39s AVGTE_ : 59min 5s AVGTF_ : 59min 26s AVGTG_ : 56min 44s AVGTH_ : 58min 25s AVGTI_ : 59min 33s AVGTJ_ : 57min

Laufzeit (<i>CPLEX</i> -Optimierung)	jeweils
	Laufzeit-Durchschnitt für ein Szenario ^{a)} ,
	Laufzeit für alle Szenarien (absoluter Wert),
	Laufzeit für alle Szenarien (Anteil an der Gesamt- laufzeit ^{b)}):
	AVGTa_: 24,88s 2488,15s 76,66 %
	AVGTb_: 25,23s 2523,37s 76,91 %
	AVGTc_: 24,72s 2471,94s 76,50 %
	AVGTd_: 24,76s 2475,69s 76,66 %
	AVGTe_: 25,31s 2531,21s 76,98 %
	AVGTf_: 25,81s 2581,49s 77,44 %
	AVGTg_: 24,25s 2425,14s 76,20 %
AVGTh_: 24,97s 2497,19s 76,72 %	
AVGTi_: 25,70s 2569,60s 77,20 %	
AVGTj_: 24,11s 2411,48s 76,08 %	
Optimierungsergebnisse	siehe Abb. 4.21
Ergebnis / Anmerkungen	
Laufzeitverhalten: Die <i>CPLEX</i> -Laufzeit (Durchschnitts- und Gesamtwert) sowie die Gesamtlaufzeit des Programms können als unabhängig vom Wahrheitswert betrachtet werden. Daher liegt auch der Anteil der <i>CPLEX</i> -Laufzeit an der Gesamtlaufzeit des Programms konstant bei ungefähr 77 %.	
Optimierungsergebnis: Die Optimierungsergebnisse sind in Abb. 4.21 zusammengestellt. Die Anzahl der über die gesamte Zuführungsdauer zugeführten Kunden nimmt mit dem Wahrheitswert monoton ab. Dieses Verhalten erklärt sich aus der Monotonie der unscharfen Kapazitätsschranken (vgl. Def. 4.2).	

- a. Der Durchschnittswert wird über alle Optimierungsläufen berechnet. Die Werte sind der Datei `AVGRx_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Szenarien und die Standardabweichung der Laufzeit.
- b. Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (siehe Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

truth value	time steps											sum
	0	1	2	3	4	5	6	7	8	9	10	
0,1	136	74	55	50	51	35	35	44	40	50	151	721
0,2	141	70	53	49	51	35	35	45	40	50	146	715
0,3	146	67	51	48	51	36	35	45	40	50	140	709
0,4	144	65	50	48	51	36	35	45	41	50	137	702
0,5	149	62	49	48	51	36	35	45	41	50	133	699
0,6	148	62	49	49	51	34	31	45	42	50	132	693
0,7	146	61	48	49	51	35	31	45	42	50	129	687
0,8	150	58	46	48	51	35	30	45	42	51	125	681
0,9	153	56	45	47	51	37	29	44	42	50	122	676
1	156	54	43	46	51	39	28	44	42	50	119	672
weighted sum												687

Abbildung 4.21: Auszug aus `AVGTx_analysis.xls`. Für die Wahrheitswerte 0,1 – 1,0 und für jeden Zeitschritt der Kundenzuführung, $t = 0, \dots, 10$ sind die optimierten Anzahlen zuzuführender Kunden sowie die Summe für die gesamte Dauer der Kundenzuführung angegeben. Unten rechts steht der mit den Wahrheitswerten gewichtete Mittelwert der Summen.

4.4.2.3 Anwendung des Lagrange-Verfahrens

Fall 5

Dieser Sonderfall des zweiten Optimierungsproblems mit stochastischen Nebenbedingungen entspricht dem ersten Optimierungsproblem mit stochastischen Nebenbedingungen (vgl. Abschnitt 3.4).

Datei-Vorsilbe	LGR1_
OPTIMIZATION_MODE	"SCENARIO_LAGRANGE"
AQUISE_DURATION	"100"
AQUISE_INPUT_DURATION	"0"
SCENARIO_NUMBER	"20"
TRUTH_LEVEL	"0.75"
Akquisedauer	75
zeitlicher Akquiseplan: Anzahl der Knoten	1377
zeitlicher Akquiseplan: Anzahl der Kanten	4812
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	1377
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	Der Wert gilt jeweils für ein Szenario. 1230
Anzahl der Kunden-Verteilungs-Nebenbedingungen	4514
Laufzeit (gesamt)	14min 31s

Laufzeit (<i>CPLEX</i> -Optimierung)	2min 48s (1. Iteration) 4min 58s (2. Iteration) 54 % der Gesamtlaufzeit
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	für $t = 0$ aufgerundet: 173
Ergebnis / Anmerkungen	
Laufzeitverhalten: (vgl. <code>LGR1_console.txt</code>) Die Rechenzeit der <i>CPLEX</i> -Optimierung macht über die Hälfte der Gesamtlaufzeit aus. Die Überführung des <i>CPLEX</i> -Modells in den <i>CPLEX</i> -Algorithmus (Methode <code>IlloAlgorithm::extract()</code>) benötigt mit etwa 79s (1. Iteration) und 92s (2. Iteration) insgesamt fast 20 % der Gesamtlaufzeit.	
Optimierungsergebnis: Für den Parameterwert <code>EPS = "0.001"</code> wird in zwei Lagrange-Iterationen eine Lösung gefunden. Der Durchschnittswert der absoluten Residuen der Kunden-Verteilungs-Nebenbedingungen beträgt 0.00002082 (<code>LGR1_aquisePlan.lgr</code>). Im Gegensatz zum Mehrgüterfluss-Problem ohne stochastische Nebenbedingungen (Fall 1) sind die Lösungswerte nicht ganzzahlig und wesentlich kleiner als in jenem Fall.	
Anmerkungen: Die Zielfunktion enthält eine Summe von $20 \cdot 4514$ quadrierten Kunden-Verteilungs-Nebenbedingungen. Die Modelldatei des Problems (Textdatei im LP-Format ^a) hat eine Größe von etwa 18MB (<code>LGR1_aquisePlan__2.lp</code>). Es wurden nur eine Modell- und eine Ergebnisdatei geschrieben (<code>LGR1_console.txt</code>). Für Optimierungsprobleme mit quadratischer Zielfunktion und zahlreichen Variablen ist das Laufzeitverhalten der C++-Schnittstelle von <i>CPLEX</i> unzureichend ^b .	

a. siehe Fußnote 31 auf Seite 170

b. Zum Vergleich: Das Lesen der *CPLEX*-Modelldatei `LGR1_aquisePlan__2.lp` mit der interaktiven Variante von *CPLEX* benötigt nur etwa 3s. Allerdings sind in dieser von *CPLEX* erzeugten LP-Datei alle Terme ausmultipliziert und zusammengefasst.

Fall 6

Der Vergleich von Fall 4 und Fall 5 zeigt folgendes:

- Das Lagrange-Verfahren benötigt für die Lösung des zweiten Optimierungsproblems ein Vielfaches der Rechenzeit, die das Mittelwert-Verfahren benötigt. Allein die Lösung der quadratischen Optimierungsprobleme in den beiden Iterationen des Lagrange-Verfahrens benötigt die 27-fache Rechenzeit aller 20 linearen Optimierungsprobleme des Mittelwert-Verfahrens zusammen.
- Die Optimierungsergebnisse sind unterschiedlich (187 gegenüber 173), besitzen jedoch die gleiche Größenordnung. Der mit dem Lagrangeverfahren ermittelte Optimalwert tritt beim Lösungsverfahren „Mittelwertbildung“ für kein Szenario auf (vgl. `AVG1_aquisePlan.sol`).
- Das Ergebnis mit dem Lagrange-Verfahren ist „sicherer“ gegenüber der Verletzung von Kapazitätsschranken als das Ergebnis mit dem Mittelwert-Verfahren.

Fall 8

Datei-Vorsilbe	LGR2a_ – LGR2g_
OPTIMIZATION_MODE	"SCENARIO_LAGRANGE"
AQUISE_DURATION	LGR2a_ – LGR2e_ : "50" ^{a)} LGR2f, LGR2g: "40" ^a
AQUISE_INPUT_DURATION	"5"
SCENARIO_NUMBER	LGR2a_ : "1" LGR2b_ : "5" LGR2c_ : "10" LGR2d_ : "15" LGR2e_ : "20" LGR2f_ : "15" LGR2g_ : "20"
TRUTH_LEVEL	"0.75"
Akquisedauer	LGR2a_ – LGR2e_ : 55 ^a LGR2f, LGR2g: 45 ^a
zeitlicher Akquiseplan: Anzahl der Knoten	LGR2a_ – LGR2e_ : 5687 LGR2f, LGR2g: 4547
zeitlicher Akquiseplan: Anzahl der Kanten	LGR2a_ – LGR2e_ : 19822 LGR2f, LGR2g: 15802
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	LGR2a_ – LGR2e_ : 5687 LGR2f, LGR2g: 4547
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	LGR2a_ – LGR2e_ : 890 LGR2f, LGR2g: 720
Anzahl der Kunden-Verteilungs-Nebenbedingungen	LGR2a_ – LGR2e_ : 18087 für jedes Szenario LGR2f, LGR2g: 14307 für jedes Szenario
Laufzeit (gesamt)	LGR2a_ : 1min 18s LGR2b_ : 4min 38s LGR2c_ : 10min 1s LGR2d_ : 14min 46s ^a LGR2e_ : --- ^a LGR2f_ : 16min 39s LGR2g_ : 17min 59s ^a

Laufzeit (<i>CPLEX</i> -Optimierung)	jeweils Laufzeit-Durchschnitt für eine Iteration ^{b)} , Laufzeit für alle Iterationen (absoluter Wert), Laufzeit für alle Iterationen (Anteil an der Gesamtlaufzeit ^{c)}): LGR2a_: 31,17s 62,35s 79,82 % LGR2b_: 112,21s 224,19s 80,69 % LGR2c_: 233,43s 466,85s 77,65 % LGR2d_: 454,75s --- ^a LGR2e_: --- ^a LGR2f_: 294,41s 588,82s 60,83 % LGR2g_: 530,62s --- ^a
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kun- den)	für t = 0, 1, ..., 5 jeweils aufgerundet sowie die Summe der aufgerundeten Werte: LGR2a_ (1 Szenario) 58, 56, 57, 58, 58, 95; Summe: 382 LGR2b_ (5 Szenarien) 72, 58, 58, 58, 60, 71; Summe: 377 LGR2c_ (10 Szenarien) 69, 57, 55, 54, 54, 81; Summe: 370 LGR2d_ (15 Szenarien) 56, 54, 54, 54, 54, 84; Summe: 356 ^a LGR2e_ (20 Szenarien) --- ^a LGR2f_ (15 Szenarien) 64, 58, 54, 54, 54, 81; Summe: 365 LGR2e_ (20 Szenarien) 56, 54, 54, 54, 54, 82; Summe: 354 ^a
Ergebnis / Anmerkungen	
Laufzeitverhalten: Der Anteil der <i>CPLEX</i> -Laufzeit an der Gesamtlaufzeit ist in den Fällen mit Aquisedauer 55 mit rund 80 % sehr hoch. Aus den <i>CPLEX</i> -Protokolldateien *.cpx ist die enorme Größe der quadratischen Optimierungsprobleme ersichtlich. Die Zielfunktion von LGR2g_ enthält 20 · 14307 quadratische Terme – entsprechend den Kunden-Verteilungs-Nebenbedingungen.	

Optimierungsergebnis: Im Gegensatz zum Mehrgüterfluss-Problem (Fall 2) sind die Lösungswerte nicht ganzzahlig und wesentlich kleiner als in jenem Fall (`LGR2a_aquisePlan.lgr` – `LGR2d_aquisePlan.lgr`).

Die Anzahl der zugeführten Kunden nimmt mit der Anzahl der Szenarien ab. Das folgt unmittelbar aus der gleichzeitigen Berücksichtigung aller Szenarien in der quadratischen Zielfunktion.

Für einige der Knoten-Kapazitäts-Nebenbedingungen finden sich in den Lösungsdateien *.sol Schlupfvariablen mit Wert gleich null. Das zeigt, dass die Anzahl der Kunden durch Kapazitätsbeschränkungen bestimmt wird.

Dieses Verhalten zeigt sich auch in den Fällen `LGR2f_`, `LGR2g_` mit verkürzter Akquisedauer.

Anmerkungen: Dem Parameter `AQUISE_DURATION` (vgl. Abschnitt 4.4.1.1) ist der Wert 50 zugewiesen. Für $t = 50$ ist die Präsenzwahrscheinlichkeit in den Endknoten 95 %:

$p_{N-1}(t) + p_N(t) = 0,95$. Tatsächlich beträgt die Akquisedauer bei einmaliger Kundenzuführung 75 Zeitschritte (= Tage). Durch die Reduzierung auf 50 ergibt sich eine Ungenauigkeit. Das Lagrangeverfahren lässt jedoch mit der verwendeten Hardware und mit einer nennenswerten Anzahl von Szenarien keine längeren Akquisedauern zu. Näheres dazu: siehe Fall 9.

Für `AQUISE_DURATION = "50"` und `SCENARIO_NUMBER = "15"` wird der Optimierungslauf mit einer Ausnahme („exception“) nach der ersten Iteration mit einem Residuum von 0,07652655 abgebrochen. Der für den Abbruch vorgegebene Parameterwert ist `AQUISE_DURATION_ACCURACY = "0.001"`. Für `AQUISE_DURATION = "50"` und `SCENARIO_NUMBER = "20"` wird der Optimierungslauf noch vor der ersten Iteration mit einer Ausnahme („exception“) abgebrochen. Diese Ausnahmen treten immer beim Aufruf der Methode `IloAlgorithm::extract(...)` der *CPLEX C++*-Schnittstelle^{d)} auf, und immer dann, wenn das Optimierungsproblem eine bestimmte Größe überschreitet. Teilweise wird dabei die Ausnahme `<IloAlgorithm cannot extract extractable>` geworfen. Ihre Bedeutung: „This message indicates that there is some problem with the model or an incompatibility between the instance of `IloAlgorithm` and the `IloModel`.“ [79] kann nicht nachvollzogen werden.

Damit das Lagrange-Verfahren mit 15 und auch mit 20 Szenarien durchgeführt werden kann, wird dem Parameter `AQUISE_DURATION` der Wert 40 zugewiesen. Für diesen Parameterwert treten die gleichen Probleme auf – aber bei höheren Szenarienzahlen: Bei 20 Szenarien wird der Optimierungslauf mit einer Ausnahme nach der ersten Iteration mit einem Residuum von 0.07674178 abgebrochen. Der für den Abbruch vorgegebene Parameterwert ist `AQUISE_DURATION_ACCURACY = "0.001"`.

a. siehe Anmerkungen

b. Der Durchschnittswert wird über alle Optimierungsläufe berechnet. Die Werte sind den Ausgabe-Dateien `LGR2x_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Iterationen und die Standardabweichung der Laufzeit.

c. Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (siehe Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

d. siehe Fußnote 26 auf Seite 168

Fall 9

Der Vergleich von Fall 7 und Fall 8 zeigt folgendes:

- Ein direkter Vergleich von AVG2_ mit LGR2x_ ist nicht möglich, weil das Lagrange-Verfahren auf den zur Verfügung stehenden Computern nicht mit 20 Szenarien durchgeführt werden kann. Für die Parameterkombination

- AQUISE_DURATION = "50",
- AQUISE_INPUT_DURATION = "5",
- SCENARIO_NUMBER = "20"

wird der Optimierungslauf mit einer Ausnahme („exception“) vor der ersten Iteration (LGR2e_) oder nach der ersten Iteration (LGR2g_) abgebrochen. Diese Ausnahme tritt immer beim Aufruf der Methode `IloAlgorithm::extract(...)` der *CPLEX C++*-Schnittstelle⁴⁷⁾ auf, und immer dann, wenn das Optimierungsproblem eine bestimmte Größe überschreitet. Teilweise wird dabei die Ausnahme `<IloAlgorithm cannot extract extractable>` geworfen. Ihre Bedeutung: „This message indicates that there is some problem with the model or an incompatibility between the instance of `IloAlgorithm` and the `IloModel`.“ [79] kann jedoch nicht nachvollzogen werden, da der Modelaufbau unabhängig von der Anzahl der Szenarien ist und bei kleineren Szenarien-Anzahlen die Ausnahme nicht geworfen wird.

Im Debug-Modus der *Microsoft Visual C++*-Entwicklungsumgebung wird als Ursache des Programmabbruchs „stack overflow“ angezeigt. Durch explizite Vorgabe einer Stack-Größe von 200 – 300MB bei der Übersetzung des C++-Programms wird das Problem entschärft, aber nicht behoben. Bei der Problemgröße LGR2e_ wird vorübergehend Arbeitsspeicher von weit über 1 GB benötigt, so dass mit der verwendeten Hardware ein Teil des Arbeitsspeicherinhalts auf die Festplatte ausgelagert wird, was bekanntermaßen die Stabilität von Programmen beeinträchtigt. Deshalb wird AVG2_ (20 Szenarien) mit

- LGR2d_ (Akquise-Dauer: 50, 15 Szenarien) und
- LGR2g_ (Akquise-Dauer: 40, 20 Szenarien)

verglichen.

Indikatoren der hohen Hardware-Anforderungen sind auch die Größen der erzeugten Modell- und Ergebnisdateien: LGR2c_aquisePlan.lp ist etwa 24MB groß und LGR2c_aquisePlan___2.sol etwa 4MB.

- Das Lagrange-Verfahren, das 15 Szenarien zugleich berücksichtigt benötigt mit über 16min wesentlich länger als die 20 linearen Optimierungsprobleme, die jeweils ein Szenario berücksichtigen, zusammen (5min 20s).
- Die Optimierungsergebnisse beider Verfahren zeigt folgende Tabelle.

Mit dem Lagrange-Verfahren ist die Summe der Kunden geringer als mit dem Mittelwert-Verfahren. Dieses Verhalten ergibt sich aus der gleichzeitigen Berücksichtigung aller Szenarien. Würde beim Mittelwert-Verfahren anstatt der Mittelwertbildung über die Anzahlen zugeführter Kunden für jeden Zeitschritt das „sicherere“ Minimum ausgewählt, dann würden mit diesem Verfahren insgesamt nur 371 Kunden zugeführt (vgl. AVG2_analysis.xls). Offensichtlich ist das Ergebnis mit dem Lagrange-Verfahren

47. siehe Fußnote 26 auf Seite 168

Verfahren	Anzahl zugeführter Kunden für $t = 0, 1, 2, 3, 4, 5$	Summe
AVG2_	69, 57, 56, 56, 55, 130	423
LGR2d_	56, 54, 54, 54, 54, 84	356
LGR2g_	56, 54, 54, 54, 54, 82	354

Tabelle 4.4: Vergleich der Anzahlen zugeführter Kunden für $t = 0, 1, \dots, 5$ für AVG2_, LGR2d_ und LGR2g_.

bezüglich der Einhaltung der Kapazitäts-Schranken „sicherer“ als das Mittelwert-Verfahren.

Fall 11 + Fall 12

Die Untersuchung des Laufzeitverhaltens für das Lagrange-Verfahren im Umfang der Untersuchung für das Mittelwert-Verfahren im Fall 10 ist nicht möglich. Die maximale Problemgröße für das Lagrange-Verfahren liegt bei 15 – 20 Szenarien und einer Dauer der Kundenzuführung von 5. Für diese Problemgrößen benötigt das Lagrange-Verfahren wesentlich länger als das Mittelwert-Verfahren.

Die „sichereren“ Ergebnisse des Lagrange-Verfahrens können auch mit einer Modifikation des Mittelwert-Verfahrens erhalten werden, indem nicht der Mittelwert der Anzahlen zugeführter Kunden für jeden Zeitschritt berechnet wird, sondern das Minimum dieser Anzahlen. Das hätte gegenüber dem Lagrange-Verfahren den Vorteil, dass mehr Szenarien und längere Dauern der Kundenzuführung betrachtet werden können.

Fall 14

Datei-Vorsilbe	LRGTa_ – LRGTj_
OPTIMIZATION_MODE	"SCENARIO_LAGRANGE"
AQUISE_DURATION	LRGTa_ – LRGTj_ : "50" ^{a)}
AQUISE_INPUT_DURATION	LRGTa_ – LRGTj_ : "5"
SCENARIO_NUMBER	LRGTa_ – LRGTj_ : "10"
TRUTH_LEVEL	LRGTa_ : "0.1" LRGTb_ : "0.2" LRGTc_ : "0.3" LRGTD_ : "0.4" LRGTd_ : "0.4" LRGTf_ : "0.6" LRGTg_ : "0.7" LRGTi_ : "0.9" LRGTj_ : "1.0"
Akquisedauer	LRGTa_ – LRGTj_ : 55
zeitlicher Akquiseplan: Anzahl der Knoten	LRGTa_ – LRGTj_ : 5687
zeitlicher Akquiseplan: Anzahl der Kanten	LRGTa_ – LRGTj_ : 19822
Anzahl der Fluss-Erhaltungs-Nebenbedingungen	LRGTa_ – LRGTj_ : 5687
Anzahl der Knoten-Kapazitäts-Nebenbedingungen	LRGTa_ – LRGTj_ : 890
Anzahl der Kunden-Verteilungs-Nebenbedingungen	Der Wert gilt jeweils für ein Szenario. LRGTa_ – LRGTj_ : 18087
Laufzeit (gesamt)	LRGTa_ : 9min 54s LRGTb_ : 9min 55s LRGTc_ : 9min 43s LRGTd_ : 10min 2s LRGTf_ : 9min 52s LRGTf_ : 9min 56s LRGTg_ : 9min 55s LRGTi_ : 9min 54s LRGTi_ : 10min 5s LRGTj_ : 10min 11s

Laufzeit (<i>CPLEX</i> -Optimierung)	jeweils Laufzeit-Durchschnitt für ein Szenario ^{b)} , Laufzeit für alle Szenarien (absoluter Wert), Laufzeit für alle Szenarien (Anteil an der Gesamtlaufzeit ^{c)}): <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>LRGTa_:</div> <div>230,65s</div> <div>461,29s</div> <div>77,58 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTb_:</div> <div>231,41s</div> <div>462,83s</div> <div>77,78 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTc_:</div> <div>225,04s</div> <div>450,09s</div> <div>77,20 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTd_:</div> <div>234,83s</div> <div>469,67s</div> <div>77,93 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTe_:</div> <div>229,66s</div> <div>459,32s</div> <div>77,51 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTf_:</div> <div>230,99s</div> <div>461,98s</div> <div>77,51 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTg_:</div> <div>231,63s</div> <div>463,26s</div> <div>77,83 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTh_:</div> <div>231,13s</div> <div>462,26s</div> <div>77,80 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTi_:</div> <div>236,37s</div> <div>472,74s</div> <div>78,21 %</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>LRGTj_:</div> <div>239,32s</div> <div>478,65s</div> <div>78,40 %</div> </div>
Optimierungsergebnisse (Anzahl der zum angegebenen Zeitpunkt zugeführten Kunden)	siehe Abb. 4.22
Ergebnis / Anmerkungen	
Laufzeitverhalten: Die <i>CPLEX</i> -Laufzeit (Durchschnitts- und Gesamtwert) sowie die Gesamtlaufzeit des Programms können als unabhängig vom Wahrheitswert betrachtet werden. Daher liegt auch der Anteil der <i>CPLEX</i> -Laufzeit an der Gesamtlaufzeit des Programms konstant bei ungefähr 78 %.	
Optimierungsergebnis: Die Optimierungsergebnisse sind in Abb. 4.22 zusammengestellt. Die Anzahl der über die gesamte Zuführungsdauer zugeführten Kunden nimmt mit dem Wahrheitswert „annähernd monoton“ ab. Dieses Verhalten erklärt sich aus der Monotonie der unscharfen Kapazitätsschranken (vgl. Def. 4.2).	
Anmerkungen:	

a. siehe Anmerkungen zum Fall 8

b. Der Durchschnittswert wird über alle Optimierungsläufe berechnet. Die Werte sind der Datei `AVGRx_console.txt` entnommen, die weitere Laufzeitinformationen enthält, darunter die Laufzeit aller Szenarien und die Standardabweichung der Laufzeit.

c. Bei dieser Berechnung ist in der Gesamtlaufzeit der Anteil für die Zerstörung der Objekte `OPTSOLVER2` und `OPTPROBLEM2` (siehe Abschnitt 4.4.1.5) aus Verfahrensgründen nicht enthalten.

Fall 15

Der Vergleich von Fall 13 und Fall 14 zeigt folgendes:

- Ein direkter Vergleich von `AVGTx_` mit `LGRTx_` ist nicht möglich, weil das Lagrange-Verfahren auf den zur Verfügung stehenden Computern nicht mit 20 Szenarien durchgeführt

truth value	time steps						sum
	0	1	2	3	4	5	
0,1	69	56	54	54	54	96	383
0,2	69	56	54	54	54	96	383
0,3	69	58	54	54	54	94	383
0,4	69	54	54	54	54	94	379
0,5	69	54	54	54	54	94	379
0,6	69	59	56	54	54	81	373
0,7	69	57	55	54	54	81	370
0,8	69	58	55	54	54	81	371
0,9	69	57	54	54	54	81	369
1	69	57	54	54	54	81	369
weighted sum							373

Abbildung 4.22: Auszug aus `LGRTx_analysis.xls`. Für die Wahrheitswerte („truth values“) 0,1 – 1,0 und für jede Kundenzuführung („time steps“), $t = 0, \dots, 5$ sind die optimierten Anzahlen zuzuführender Kunden sowie die Summe für die gesamte Dauer der Kundenzuführung angegeben. Unten rechts steht der mit den Wahrheitswerten gewichtete Mittelwert der Summen.

werden kann; Näheres dazu: siehe Fälle 9, 11 und 12 sowie die Anmerkungen zum Fall 8. Deshalb wird `AVGTx_` (20 Szenarien, Dauer des Zuführungszeitraums: 10) mit `LGRTx_` (10 Szenarien, Dauer des Zuführungszeitraums: 5) verglichen.

- Mit beiden Verfahren nimmt die Anzahl der über die gesamte Zuführungsdauer zugeführten Kunden mit dem Wahrheitswert monoton ab. Dieses Verhalten erklärt sich aus der Monotonie der unscharfen Kapazitätsschranken (vgl. Def. 4.2).
- Mit dem Lagrange-Verfahren beträgt die Differenz der zugeführten Kunden zwischen den Werten für Wahrheitswert 0,1 und Wahrheitswert 1,0 etwa 3,7 % des Wertes für Wahrheitswert 0,1. Mit dem Mittelwert-Verfahren beträgt diese Differenz etwa 13,7 % des entsprechenden Wertes für Wahrheitswert 0,1. Das Optimierungsergebnis mit dem Mittelwert-Verfahren hängt somit stärker von der Wahl des Wahrheitswertes ab als das mit dem Lagrange-Verfahren berechnete Ergebnis.

5 Mathematische Modellierung dynamischer Akquisepläne

Im Kapitel 3 wurde die modellbasierte Neukundengewinnung mit Akquiseplänen eingeführt. Das Modell ist die Grundlage zur Simulation des zeitlichen Verlaufs einer Akquise (siehe Abschnitt 3.3) und für die Beschreibung von drei unscharfen stochastischen Optimierungsproblemen (siehe Abschnitt 3.4). In diesem Kapitel werden die Stärken und Schwächen von Akquiseplänen untersucht und davon ausgehend eine Erweiterung vorgestellt. Die erweiterten Akquisepläne sind nur ein Zwischenschritt zu den so genannten dynamischen Akquiseplänen. Sie ermöglichen die Formulierung weiterer interessanter Optimierungsprobleme, die zum Bereich der unscharfen stochastischen dynamischen Optimierung gehören und im Kapitel 6 behandelt werden. Die Grundlagen dynamischer Akquisepläne sind die Arbeiten [161, 162, 163, 164].

5.1 Stärken und Schwächen der Kundengewinnung mit Akquiseplänen

Die wichtigste Frage für die Durchführung einer Akquise ist die nach der Wahl der richtigen Aktionen. Die im Abschnitt 3.2 eingeführten Akquisepläne legen die Aktionen in Abhängigkeit vom Kundenverhalten fest. Der Akquiseplan kann nur ganz, im Vergleich mit anderen Akquiseplänen, bewertet werden. Der Vergleich zweier unterschiedlicher Akquisepläne ist nur dann möglich, wenn die mit ihnen bearbeiteten Zielgruppen vergleichbar sind. Aktionen eines Akquiseplans können untereinander nur im Kontext des Akquiseplans verglichen werden. Eine Optimierung von Akquiseplänen unter dem Gesichtspunkt der Gewinnmaximierung oder der Minimierung oder Maximierung einer anderen ökonomischen Größe ist somit praktisch nur durch vergleichende Analysen von simulierten Akquisen möglich.

Die Anwendung von Akquiseplänen auf eine undifferenzierte, wenig bekannte Kundengruppe ist trotz der Berücksichtigung der Kundenreaktion innerhalb eines groben Rasters relativ starr. Sie entspricht weder dem Vorgehen des traditionellen Verkäufers oder Vertreters, der „eine Nase“ für die richtige Kundenansprache hatte noch dem heute propagierten Marketing-Ideal der „Eins-zu-Eins-Kundenbeziehung“ („one-to-one customer relationship“) [15], das unter Einsatz so genannter „intelligenter“ Verfahren verfolgt wird, wozu z. B. Datenmustererkennung („data mining“), Entscheidungsunterstützungssysteme („decision support systems“) [193, 194] und Agenten-Techniken („agent technologies“) [51, 52] gehören.

Eine weitere Schwäche von Akquiseplänen ist ihre eingeschränkte Berücksichtigung der Besonderheiten des einzelnen Kunden. Die Wahl einer Aktion hängt nur von der Reaktion des Kunden ab, der Ansatz sieht die Einbeziehung von Kundeneigenschaften nicht vor. In der Praxis wird ein Akquiseplan auf eine Zielgruppe zugeschnitten sein, die durch Kundensegmentierung aus einer größeren Gruppe gewonnen werden kann (siehe Abb. 5.1). Dieses Vorgehen setzt aber voraus, dass ausreichend Kundendaten für eine Segmentierung vorhanden sind, wie z. B. bei der Bestandskundenwerbung. Bei der Neukundenwerbung ist das aber nicht immer der Fall oder die vorherige Besorgung relevanter Kundendaten, z. B. über so genannte „Listbroker“ ist mit zu hohen Kosten verbunden.

Oft werden die für eine Kundensegmentierung verwertbaren Informationen über die Kunden erst im Verlauf der Akquise gewonnen. Solche Informationen bleiben bei einer auf einem Akquiseplan beruhenden Akquise unberücksichtigt, obwohl daraus wichtige Folgerungen für die Wahl der Akquiseaktionen gezogen werden könnten¹⁾. Diese Unzulänglichkeit kann durch

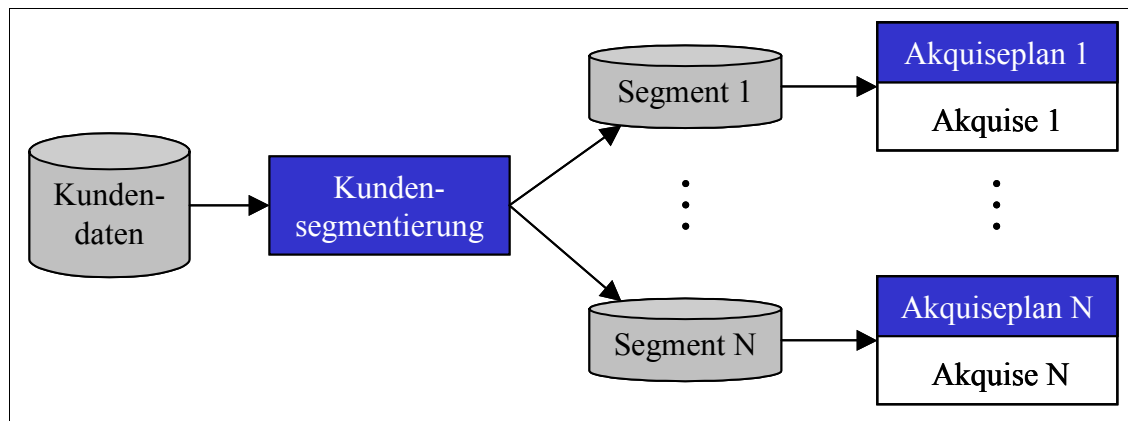


Abbildung 5.1: Der Akquise vorgeschaltete Kundensegmentierung.

Differenzierung nach Kundensegmenten im Akquiseplan behoben werden. Die Kundensegmentierung geht der Akquise nicht voraus, sondern ist ihr Bestandteil. Ein solchermaßen veränderter Akquiseplan wird als *erweiterter Akquiseplan* bezeichnet (siehe Abb. 5.2).

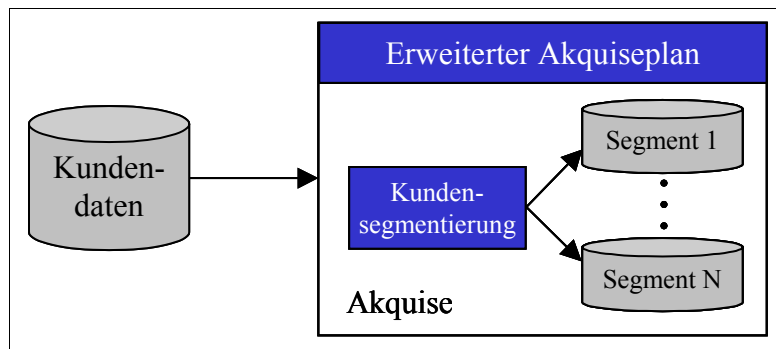


Abbildung 5.2: Erweiterter Akquiseplan mit einer in der Akquise integrierten Kundensegmentierung.

Er wird im folgenden Abschnitt erläutert und daraus im Abschnitt 5.3 der dynamische Akquiseplan entwickelt, der die Kundensegmentierung so integriert, dass die automatische Optimierung der Akquiseaktionen möglich ist.

5.2 Erweiterte Akquisepläne

In den Akquiseplänen des Software-Werkzeugs *AkquiSys* beziehen sich Kundenreaktionen auf die Haltung des Kunden zum angebotenen Produkt bzw. zur angebotenen Dienstleistung. In dem in Abb. 3.1 gezeigten Beispiel aus dem *AkquiSys*-Handbuch sind folgende direkte Kundenreaktionen enthalten:

- Rückruf erbeten,
- kein Interesse,

1. Zur Abhängigkeit der Wahl der richtigen Marketing-Maßnahmen von Kundenmerkmalen vgl. z. B. [89, 92, 98].

- möchte kaufen,
- ist einverstanden

sowie die indirekten Kundenreaktionen:

- kein Interesse,
- Bedenkzeit,
- Angebot abgelehnt.

Der Akquiseplan enthält in Gestalt seiner Kanten alle möglichen Kundenreaktionen oder, anders gesehen, das mögliche Produktinteresse der Kunden ist im Akquiseplan „gespeichert“. In Abhängigkeit von den einzelnen Akquiseaktionen können aus der Kundenreaktion weitere Kundeneigenschaften außer dem Produktinteresse gewonnen werden. Wenn auch diese Informationen durch Kanten dargestellt werden, dann geht der Akquiseplan in einen erweiterten Akquiseplan über. Für diese Erweiterung ist zunächst festzulegen, welche Kundenmerkmale festgehalten werden sollen. Möglich sind z. B.:

- Einkommen,
- Alter,
- Bildungsstand,
- Produktinteresse.

Das Kundenmerkmal „Produktinteresse“ könnte beispielsweise Werte annehmen, die den oben genannten ursprünglichen Kundenreaktionen entsprechen. Für die hinzugekommenen Kundenmerkmale sind Wertebereiche festzulegen und, um die Anzahl der Kundenreaktionen zu beschränken, eine Partitionierung der Wertebereiche. Zur Veranschaulichung zeigt Abb. 5.3 einen Ausschnitt aus einem entsprechend erweiterten Akquiseplan. Die Akquiseaktion A_i gehört zum Akquiseplan einer Versicherungsagentur und besteht aus einem Brief mit Produktinformationen und einem Antwort-Coupon, mit dem wahlweise weitere Informationen oder ein konkretes Angebot angefordert werden können. Für die Anforderung eines Angebots muss der Kunde sein Einkommen angeben. Es wird angenommen, dass von den Kunden, die mit der Aktion A_i bearbeitet werden, bereits das ungefähre Alter und der Bildungsstand bekannt sei (in der Abbildung türkis gekennzeichnet), jedoch keine Information über das Einkommen und das Produktinteresse vorliegt. Weiter wird angenommen, dass jeder Kunde auf die Aktion A_i in einer von drei möglichen Weisen reagiert:

1. Der Kunde fordert mit dem Antwort-Coupon weitere Informationen an. Nach dieser Reaktion kann sein Produktinteresse als „interessiert“ eingestuft werden. Sein Einkommen bleibt weiter „unbekannt“ (in der Abbildung grün gekennzeichnet).
2. Der Kunde fordert mit dem Antwort-Coupon ein konkretes Angebot an und gibt sein Einkommen an. Nach dieser Reaktion wird sein Produktinteresse als „Angebotswunsch“ eingestuft und sein Einkommen als „gering“, „mittelhoch“ oder „hoch“ (in der Abbildung blau gekennzeichnet). Hier könnte auch ein konkreter Betrag stehen. Dadurch würde jedoch die Zahl möglicher Reaktionen unmaßig vergrößert und der Akquiseplan unübersichtlich. Mög-

lich wäre auch eine andere Partition des Einkommens. Die konkrete Segmentierung wird von der Differenzierung der Folgeaktionen entsprechend des Einkommens bestimmt.

3. Der Kunde reagiert nicht. Nach einer festgelegten Zeit wird dies als (indirekte) „Reaktion“ bewertet. Ein unmittelbarer Informationsgewinn ist nicht möglich, das Produktinteresse bleibt „unbekannt“ und die anderen Kundeneigenschaften ändern sich ebenfalls nicht (in der Abbildung gelb gekennzeichnet). Mittelbar könnte aus dem nicht-Reagieren des Kunden, vor allem, wenn er auch auf vergangene Akquiseaktionen nicht reagiert hat, ebenfalls auf das Produktinteresse „uninteressiert“ geschlossen werden. Oder bei entsprechender Vorgeschichte und optimistischer Sichtweise auf das Produktinteresse „unentschlossen“.

Im Beispiel wird an der zweiten Art der Kundenreaktion der Unterschied zu den Akquiseplänen aus Kapitel 3 deutlich. Ein einfacher Akquiseplan würde sich nur nach

- der Anforderung weiterer Informationen,
- der Anforderung eines konkreten Angebots,
- keine Reaktion

unterscheiden und insbesondere die Höhe des Einkommens im Fall 2 nicht berücksichtigen und dementsprechend nicht mehr als drei Folgeaktionen vorsehen.

Kundensegmentierung bezweckt den wirksamen Einsatz von Marketing-Maßnahmen durch Anpassung der Maßnahmen an die Besonderheiten der jeweiligen Kundengruppe [89, 92, 97, 98, 101, 111, 151, 157]. Kundensegmentierung umfasst sowohl die Markterfassungs- bzw. Informationsseite als auch die Marktbearbeitungsseite [111]. Dem entspricht die hier eingeführte Erweiterung von Akquiseplänen. Insofern ist die Festlegung von Kundenmerkmalen zur Informationserfassung und die Partitionierung des Wertebereichs der Kundenmerkmale sowie die Zuordnung von Kunden zu einer bestimmten Reaktion eine Kundensegmentierung.

Erweiterte Akquisepläne unterscheiden sich nicht grundsätzlich von einfachen Akquiseplänen gemäß Def. 3.1. Sie besitzen die gleichen Struktureigenschaften, lassen sich in gleicher Weise aufbauen und analysieren, sind aber je nach Anzahl der betrachteten Kundenmerkmale komplexer und aufwändiger zu erstellen. Der Quellknoten eines erweiterten Akquiseplans bezeichnet, wie der Quellknoten eines einfachen Akquiseplans, die zeitlich erste Aktion der Akquise. Die von ihm ausgehenden Kanten basieren auf der zum Akquisebeginn vorliegenden Information über alle in die Akquise einbezogenen Kunden.

Eine Kundensegmentierung, die für einige Kundenmerkmale vor Akquisebeginn durchgeführt wird, kann in die Darstellung des erweiterten Akquiseplans formal einbezogen werden. Dazu wird die Kundensegmentierung als erste Aktion der Akquise betrachtet und entsprechend auf den Quellknoten abgebildet. Die auftretenden Kombinationen von Werten der Merkmale dieser Segmentierung werden als Reaktionen der Kunden betrachtet. Ihnen entsprechen die vom Quellknoten ausgehenden Kanten.

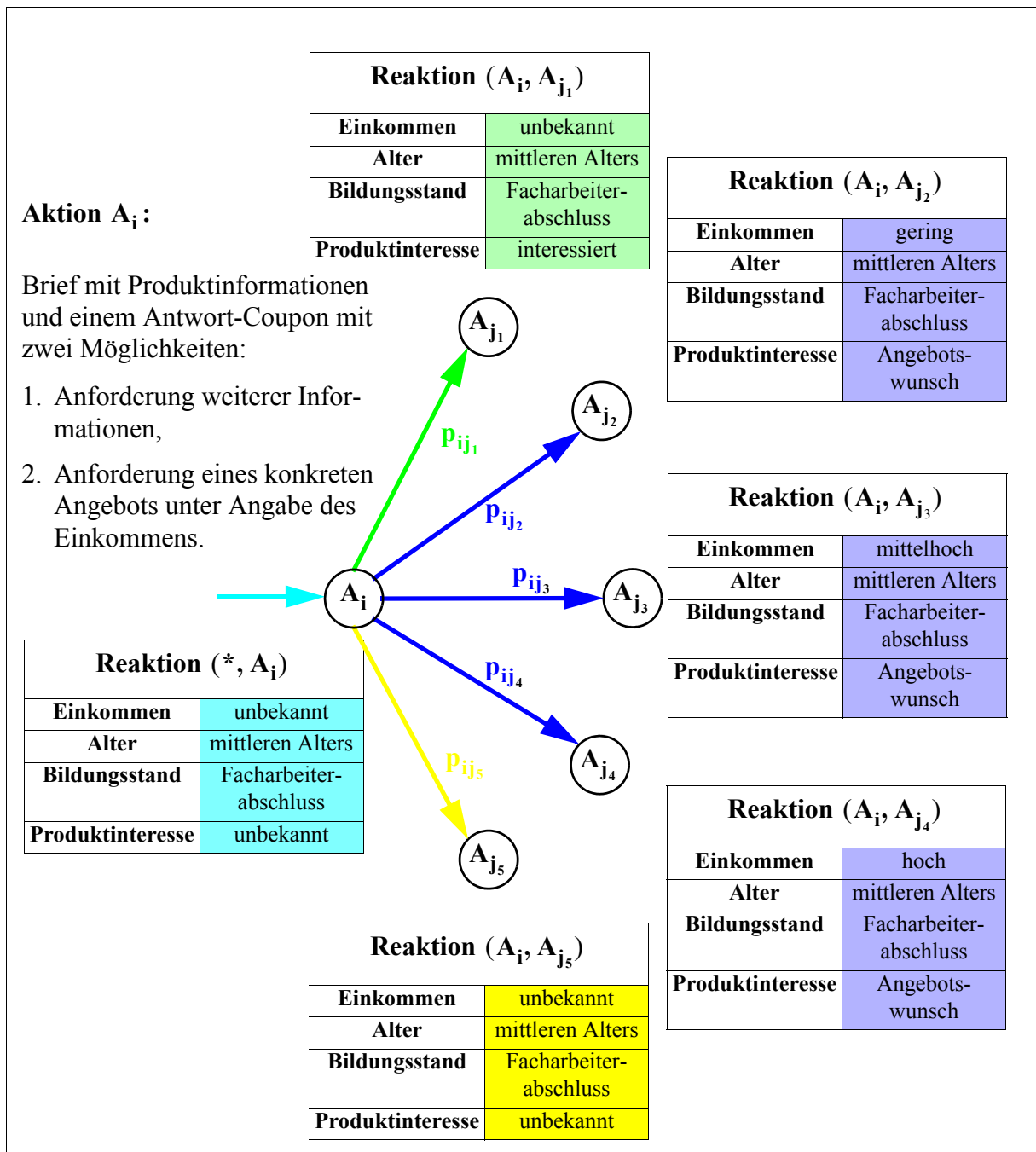


Abbildung 5.3: Beispiel: Ausschnitt aus einem *erweiterten Akquiseplan*. Die Kundenreaktionen umfassen jetzt auch Informationen über den Kunden, die durch die Akquiseaktion gewonnen werden, hier: Einkommen, Alter und Bildungsstand.

5.3 Dynamische Akquisepläne

Eine grundsätzlich neue Darstellung für Kundenakquisen wird in diesem Abschnitt eingeführt. Ihre Motivation sind die unzulänglichen Möglichkeiten des einfachen und des erweiterten Akquiseplans.

5.3.1 Notwendigkeit dynamischer Akquisepläne

Eine Erweiterung von Akquiseplänen wie im Abschnitt 5.2 durch Erweiterung der Kundenreaktion um Kundenmerkmale, die über sein Produktinteresse hinausgehen, löst die Hauptschwierigkeit von Akquiseplänen nicht:

- Zunächst muss zu jeder Kundenreaktion eine Akquiseaktion festgelegt werden.
- Die Bewertung dieser Festlegung ist nur durch die vergleichende Auswertung von unterschiedlichen Akquiseplänen, die in den einer Kundenreaktion folgenden Akquiseaktionen variieren, möglich.

Besser wäre eine automatische Berechnung der besten Akquiseaktionen. In diesem Abschnitt wird ein Modell eingeführt, das die Grundidee aus Abschnitt 5.2 aufgreift und zugleich die algorithmische Optimierung der Akquiseaktionen ermöglicht.

In [130, 131] wird die Optimierung der Massenbrief-Aktionen einer Wohltätigkeitsorganisation beschrieben. Dafür wird das Direkt-Marketing der Wohltätigkeitsorganisation, also die Bittbriefe an mögliche Neuspender und an Altspender, als dynamisches System mit einem Zustandsraum modelliert, der durch die RFM-Variablen, „recency“, „frequency“ und „monetary value“ gegeben ist und daher drei Dimensionen besitzt:

1. Die erste Dimension ist die *Neuheit* („recency“), die angibt, wie lange es her ist, seit der Kunde das letzte Mal geantwortet hat. Sie wird gemessen in Aktionen seit der letzten Antwort.
2. Die zweite ist die *Häufigkeit* („frequency“), die ein Maß bereitstellt, wie oft der Kunde überhaupt auf erhaltene Massensendungen geantwortet hat.
3. Und drittens misst der *Geldwert* („monetary value“) den Geldbetrag, den der Kunde als Antwort auf Massensendungen gespendet bzw. die Anzahl von Produkten, die er gekauft hat.

Eine anschauliche Erläuterung gibt Bsp. 5.5 in Tab. 5.4.

Die Optimierungsaufgabe besteht darin, die Anzahl der Massensendungen, die in einem Jahr an die Spender verschickt werden, so festzulegen, dass das Spendenaufkommen maximal wird. Dazu wird eine *optimale Politik* berechnet, das ist eine Entscheidungsregel, die jedem Zustand die Anzahl der Massensendungen zuordnet. Sie bestimmt für jedes Spendersegment die Anzahl der Bittbriefe im nächsten Jahr. Die Übergangswahrscheinlichkeiten zwischen den Zuständen werden aus historischen Daten geschätzt.

Ein ähnlicher Ansatz wird für die hier einzuführenden dynamischen Akquisepläne gewählt, indem der Akquiseprozess als *mehrstufiger stochastischer Entscheidungsprozess mit implizit vorgegebener Prozess-Endzeit* modelliert wird. Der neue Ansatz vermeidet die weiter oben beschriebenen Unzulänglichkeiten. Das genaue Modell wird im Kapitel 6 ausführlich behandelt. Im vorliegenden Kapitel wird das Modell zunächst aus den Anforderungen der Akquise hergeleitet. Dazu müssen einige Begriffe eingeführt werden, auf die Kapitel 6 zurückkommt bzw. die dort allgemeiner behandelt werden. Der nächste Abschnitt ist dennoch so ausführlich, dass ein Optimierungsproblem für dynamische Akquisepläne im Abschnitt 5.4 umgangs-

sprachlich formuliert werden kann – ähnlich allgemein wie die drei Optimierungsprobleme im Abschnitt 3.4.

5.3.2 Definition dynamischer Akquisepläne

Kapitel 6 führt dynamische Akquisepläne formal ein – unter der Bezeichnung „Mehrstufige stochastische Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit“. Unabhängig davon werden dynamische Akquisepläne in diesem Abschnitt mit dem Anwendungshintergrund der modellbasierten Neukundengewinnung eingeführt. Dieses Vorgehen unterstützt das bessere Verständnis des Modells. Einerseits wird das Modell in diesem Abschnitt aus den Anforderungen der Neukundengewinnung motiviert und vor allem illustriert. Abstraktion wird hier nur soweit getrieben, als sie für die Klärung der Begrifflichkeiten notwendig ist. Andererseits geht Kapitel 6 von abstrakten Begriffen ohne Anwendungshintergrund aus, wodurch die Konzentration auf Struktureigenschaften und mathematische Zusammenhänge erleichtert wird und die Lösung des Optimierungsproblems unbeschwert von unnötigem Anwendungsbalast hergeleitet werden kann.

Die folgenden Definitionen gehen von der Neukundengewinnung als zeitlich verlaufendem Prozess aus, der verkürzt als „Akquise von Kunden“ bezeichnet wird.

Definition 5.1: *Merkmalsvariable*²⁾, *Merkmalswert*, *Merkmalsmenge*, *Kundendatensatz-Variable*, *Kundendatensatz*, *Kundendatensatz-Menge*, *Standardwert*, *negKE*, *posKE*.

Eine *Merkmalsvariable* ξ ist die Variable eines Kundenmerkmals. Sie kann die *Merkmalswerte* s_1, s_2, \dots, s_n annehmen, die zusammen die *Merkmalsmenge* S des Merkmals bilden. Eine *Kundendatensatz-Variable* x besteht aus einem Tupel von Merkmalsvariablen ξ_j ,

$$x = (\xi_1, \xi_2, \dots, \xi_p),$$

wobei die Merkmalsvariable ξ_j Werte aus der endlichen Merkmalsmenge

$$S_j = \{s_{j1}, s_{j2}, \dots, s_{jn_j}\}$$

mit $j = 1, \dots, p$ annehmen kann. Ein *Kundendatensatz* σ ist ein Wert einer Kundendatensatz-Variablen,

$$\sigma = (s_1, s_2, \dots, s_p), s_j \in S_j \text{ für } j = 1, \dots, p.$$

Die *Kundendatensatz-Menge* ist die Menge aller Kundendatensätze und wird mit

$$X = \{\sigma_1, \dots, \sigma_n\}$$

bezeichnet, wobei

2. „Merkmal“ und „Eigenschaft“ werden gleichbedeutend gebraucht.

$$\sigma_i = \{s_{1i_1}, s_{2i_2}, \dots, s_{pi_p}\} \text{ für } i = 1, \dots, n \text{ mit } n = \prod_{j=1}^p n_j.$$

Für jede Merkmalsvariable sei „unbekannt“ der *Standardwert* („default value“), das heißt $\forall j = 1, \dots, p: s_{j1} = \text{unbekannt}$. Weiterhin wird o. B. d. A. angenommen, dass die Merkmalsmenge S_p Angaben zum Interesse des Kunden am Produkt enthält und darin die Merkmale „endgültige negative Kaufentscheidung“ (*negKE*) und „endgültige positive Kaufentscheidung“ (*posKE*) enthalten seien, das heißt $s_{p2} = \text{negKE}$, $s_{pn_p} = \text{posKE}$ und $s_{p1} = \text{unbekannt}$. \square

Definition 5.2: *Akquise-Zustandsmenge, Akquise-Zustand (zum Zeitpunkt t), Akquise-Zustandsvariable.*

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Kundendatensatz-Menge gemäß Def. 5.1. Dann ist die *Akquise-Zustandsmenge* gleich der Kundendatensatz-Menge $X = \{\sigma_1, \dots, \sigma_n\}$ und ein *Akquise-Zustand* gleich einem Datensatz σ_i , $i = 1, \dots, n$. Die Kundendatensatz-Variable x ist dann eine *Akquise-Zustandsvariable* und für festgelegte diskrete Zeitpunkte $t = 0, 1, \dots$ bezeichnet $x_t \in X$ den *Akquise-Zustand zum Zeitpunkt t* . \square

Anmerkungen

1. Die Zustandsmenge ist gleich dem kartesischen Produkt der Merkmalsmengen

$$X = S_1 \times \dots \times S_p. \quad (5.1)$$

2. Kundenmerkmale sind z. B. Einkommen, Alter, Bildungsstand.
3. Aus Gründen der praktischen Handhabbarkeit werden die Merkmalsmengen S_j , $j = 1, \dots, p$ eher eine geringe Anzahl von Elementen enthalten, das heißt die Granularität³⁾ der entsprechenden Information ist gering [188]. Die Werte der Merkmalsmenge Einkommen sind z. B. „gering“, „mittel“ oder „hoch“, die Merkmalsvariable für Alter nimmt z. B. die Werte „Teenager“, „jung“, „mittleren Alters“ oder „alt“ an.

Beispiel 5.1: Tabelle 5.1 zeigt einige Zustände (Kundendatensätze) in einem Zustandsraum (Kundendatensatz-Menge) mit den Merkmalsvariablen $\xi_1 = \text{Einkommen}$, $\xi_2 = \text{Alter}$, $\xi_3 = \text{Bildungsstand}$ und $\xi_4 = \text{Produktinteresse}$ und den Merkmalsmengen

- $S_1 = \{\text{unbekannt, gering, mittel, hoch}\},$
- $S_2 = \{\text{unbekannt, Teenager, jung, mittleren Alters, alt}\},$
- $S_3 = \{\text{unbekannt, Facharbeiterabschluss, Abitur, Universitätsabschluss}\},$

3. vgl. auch Abschnitt 2.5

- $S_4 = \{\text{unbekannt, negKE, unentschlossen, groß, Angebotswunsch, posKE}\}$.

	ξ_1 (Einkommen)	ξ_2 (Alter)	ξ_3 (Bildung)	ξ_4 (Interesse)
σ_1	gering	Teenager	unbekannt	groß
σ_2	mittel	jung	Facharbeiter-abschluss	unentschlossen
σ_3	unbekannt	mittleren Alters	Abitur	Angebotswunsch
σ_4	hoch	alt	Universitäts-abschluss	posKE ^a
σ_5	mittel	mittleren Alters	Facharbeiter-abschluss	negKE ^a

Tabelle 5.1: Einige Zustände in einem Zustandsraum.

a. posKE = **positive Kauf-Entscheidung**, negKE = **negative Kauf-Entscheidung** (vgl. Def. 5.1)

Beispiel 5.2: Abb. 5.4 zeigt eine Teilmenge der Zustandsmenge $\tilde{X} = S_1 \times S_2 \times S_4$ mit den Merkmalsmengen aus Bsp. 5.1.

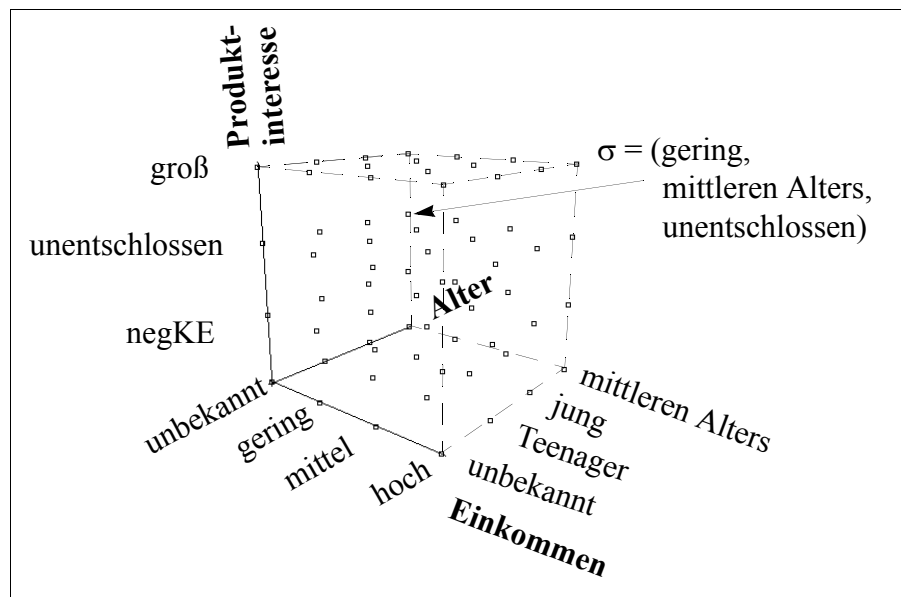


Abbildung 5.4: Beispiel einer Zustandsmenge.

Anmerkung

Die Beispiele 5.1 und 5.2 zeigen anschaulich, wie Akquise-Zustände

$$\sigma_i = \{s_{1i_1}, s_{2i_2}, \dots, s_{pi_p}\} \text{ für } i = 1, \dots, n$$

kodiert werden können: Nach Festlegung einer Ordnung der Elemente der Merkmalsmengen

$$S_j = \{s_{j1}, s_{j2}, \dots, s_{jn_j}\} \text{ für } j = 1, \dots, p$$

ist

$$\sigma_i \leftrightarrow (i_1, \dots, i_p) \text{ für } i = 1, \dots, n$$

eine bijektive Abbildung zwischen der Akquise-Zustandsmenge X und der Menge der Indizes der Merkmalswerte

$$I = \{(i_1, \dots, i_p) \mid i_j \in \{1, 2, \dots, n_j\}, j = 1, \dots, p\}.$$

Definition 5.3: *Akquise-Entscheidungsmenge, Akquise-Entscheidungsvariable.*

Die *Akquise-Entscheidungsmenge* U ist eine endliche Menge von Akquiseaktionen $\alpha_1, \dots, \alpha_m$. Mit u wird die *Akquise-Entscheidungsvariable* bezeichnet. \square

Anmerkungen

1. Unter Akquiseaktionen, kurz: Aktionen, werden alle Maßnahmen verstanden, die
 - Informationen über Produkte – Güter oder Dienstleistungen – an den Kunden weitergeben,
 - den Kunden direkt oder indirekt auffordern, auf für den Akquisiteur vorteilhafte Weise zu reagieren, z. B. Anforderung weiterer Produktinformationen, Anforderung eines Angebots, Kauf eines Produkts oder Bestellung einer Dienstleistung.
 Beispiele für solche Aktionen sind Massenbriefsendungen, Briefe, Telefonate, Produktpräsentationen, Anzeigen usw.
2. Da die Wirkung von Akquiseaktionen von Form *und* Inhalt abhängt, sind zwei Aktionen unterschiedlich, wenn sie sich in Form *oder* Inhalt unterscheiden. Beispielsweise sind ein Brief mit allgemeinen Produktinformationen und ein Brief mit Detailinformationen verschiedene Aktionen. Auch eine schriftlich erteilte Auskunft und eine telefonisch erteilte Auskunft werden als unterschiedliche Aktionen betrachtet.

Definition 5.4: *Akquise-Zustandsübergang, Akquise-Übergangswahrscheinlichkeit.*

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Akquise-Zustandsmenge X gemäß Def. 5.2. Der *Akquise-Zustandsübergang* ist durch die stochastischen Matrizen $P(\alpha)$, $\alpha \in U$ mit den *Akquise-Übergangswahrscheinlichkeiten* $p_{ij}(\alpha) = p(\sigma_j \mid \sigma_i, \alpha)$, $i, j = 1, \dots, n$ gegeben. Das Element $p_{ij}(\alpha)$ bezeichnet die Wahrscheinlichkeit, mit der der Akquise-Zustand (das heißt der Kundendatensatz) zum Zeitpunkt t , $x_t = \sigma_i$ unter der Akquiseaktion $u_t = \alpha$ in den Akquise-Zustand (Kundendatensatz) zum Zeitpunkt $t+1$, $x_{t+1} = \sigma_j$ übergeht. \square

Anmerkungen

1. Da $P(\alpha)$ für jedes $\alpha \in U$ eine stochastische Matrix ist, besitzen die Akquise-Übergangswahrscheinlichkeiten für jedes $\alpha \in U$ die Eigenschaft: $\sum_{j=1}^n p_{ij}(\alpha) = 1$.
2. Die Akquise-Übergangswahrscheinlichkeiten hängen von der Vergangenheit nur durch den aktuellen Zustand und die gewählte Aktion des Entscheiders ab⁴⁾.
3. Die Akquise-Übergangswahrscheinlichkeiten hängen nicht von der Zeit ab, sie sind stationär.
4. Die Anzahl der Übergangswahrscheinlichkeiten beträgt höchstens

$$|\{(x_t, x_{t+1}, u_t) \mid x_t, x_{t+1} \in X, u_t \in U\}| = |X|^2 \cdot |U|.$$

Das ist gleich der maximalen Anzahl an Wahrscheinlichkeiten von Knotenübergängen bei erweiterten Akquiseplänen (Abschnitt 5.2).

Beispiel 5.3: Das in Abb. 5.5 illustrierte Beispiel basiert auf der Kundendatensatz-Menge von Bsp. 5.1 und überführt den im Abschnitt 5.2 in der Abb. 5.3 gezeigten Ausschnitt eines erweiterten Akquiseplans in einen dynamischen Akquiseplan.

Sei der aktuelle Zustand x_t gleich

$$\sigma_i = (\text{unbekannt, mittleren Alters, Facharbeiterabschluss, unbekannt})$$

(Zustand in türkis) und die Entscheidung α_j ein Brief mit Produktinformationen und einem Antwort-Coupon (in rot), auf dem der Kunde auswählen kann, ob er entweder weitere Informationen oder ein konkretes Angebot wünscht. Im zweiten Fall muss der Kunde sein Einkommen angeben, damit das gewünschte Angebot seinen Verhältnissen angepasst werden kann (z. B. bei einer Lebensversicherung). Dann gibt es drei Möglichkeiten, wie der Kunde reagieren kann:

1. Er schickt den Antwort-Coupon zurück und bittet um weitere Informationen. Dann bleibt sein Einkommen weiterhin „unbekannt“ aber sein bisher unbekanntes Produktinteresse ändert sich zu „interessiert“ (Zustand σ_{j_1} , in grün).
2. Er verlangt ein Angebot und gibt sein Einkommen an. Dadurch gelangt er in Akquise-Zustände, bei welchen die Merkmalsvariable „Produktinteresse“ den Wert „Angebotswunsch“ und die Merkmalsvariable „Einkommen“ einen der Werte „gering“ (Zustand σ_{j_2}), „mittelhoch“ (Zustand σ_{j_3}) oder „hoch“ (Zustand σ_{j_4}) besitzen (alle Zustände sind blau eingezeichnet). Die Anzahl der Folgezustände bei diesem Verhalten des Kunden hängt von der Partitionierung der Merkmalsvariablen „Einkommen“ ab, das heißt von der Anzahl der Elemente in der Merkmalsmenge der Variablen.

4. Diese Eigenschaft wird auch als Markow-Eigenschaft der Übergangswahrscheinlichkeiten bezeichnet [134].

3. Er antwortet überhaupt nicht. Dann tritt Zustand σ_{j_5} ein (in gelb), wo Einkommen und Produktinteresse immer noch „unbekannt“ sind. Letzteres hängt von der Einschätzung des Akquisiteurs ab. Mittelbar könnte aus dem nicht-Reagieren des Kunden, vor allem, wenn er auch auf vergangene Akquiseaktionen nicht reagiert hat, auch auf sein Desinteresse geschlossen werden. Oder bei entsprechender Vorgeschichte und optimistischer Sichtweise auf das Produktinteresse „unentschlossen“.

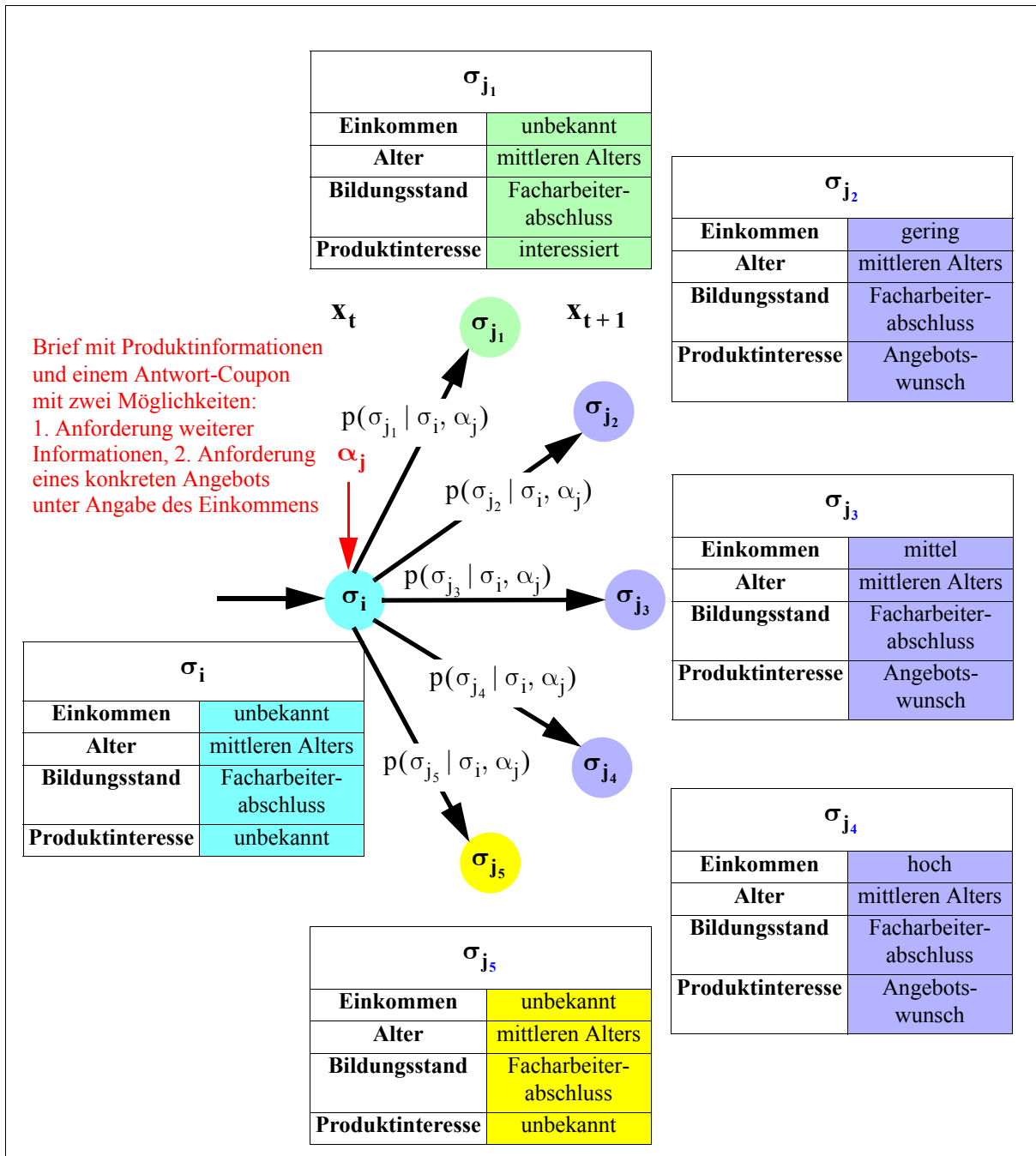


Abbildung 5.5: Beispiel möglicher Zustandsübergänge entsprechend der Kundenreaktionen in einem *dynamischen Akquiseplan*.

Die Übergangswahrscheinlichkeiten dieser drei Fälle sind $p(\sigma_{k_\nu} | \sigma_i, \alpha_j)$, $\nu = 1, \dots, 5$. Sie können auf unterschiedliche Weise gewonnen werden, z. B. durch eine systematische Analyse des Kundenantwortverhaltens oder aus den Erfahrungswerten („Daumenregeln“) von Experten.

Letztendlich verfolgt die Kundenakquisition den Zweck, einen Gewinn zu erzielen. Es wird nur dann eintreffen, wenn ein Vertragsabschluss zustande kommt. In diesem Fall ist $\xi_p = s_{pn_p} = \text{posKE}$. Gemeinsam mit dem gegensätzlichen Fall $\xi_p = s_{p2} = \text{negKE}$ ergibt sich daraus die nächste Definition:

Definition 5.5: Menge der positiven Akquise-Endzustände, Menge der negativen Akquise-Endzustände, Menge der Akquise-Endzustände, Menge der nicht-beendeten Akquise-Zustände.

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Akquise-Zustandsmenge X gemäß Def. 5.2. Die Menge der positiven Akquise-Endzustände T_{pos} ist die Menge der Zustände $\sigma_i = (s_{1i_1}, s_{2i_2}, \dots, s_{pi_p}) \in X$, $i = 1, \dots, n$ mit $s_{pi_p} = \text{posKE}$. Die Menge der negativen Akquise-Endzustände T_{neg} ist die entsprechende Menge der $\sigma_i \in X$ mit $s_{pi_p} = \text{negKE}$. Die Vereinigung $T = T_{\text{pos}} \cup T_{\text{neg}}$ bildet die Menge der Akquise-Endzustände. Jeder Akquise-Endzustand $x_t \in X$ zeichnet sich ferner vor anderen Zuständen dadurch aus, dass es keinen Folgezustand $x_{t+1} \in X$ mit $x_t \neq x_{t+1}$ gibt. $\bar{T} = X \setminus T$ ist die Menge der nicht-beendeten Akquise-Zustände. \square

Anmerkungen

1. Aus der Eigenschaft von Akquise-Endzuständen $x_t \in X$, dass es keinen Folgezustand $x_{t+1} \in X$ mit $x_t \neq x_{t+1}$ gibt, folgt, dass für die Akquise-Übergangswahrscheinlichkeiten gilt:

$$p(\sigma_j | \sigma_i, \alpha) = \delta_{ij} \text{ für alle } \alpha \in U, \quad (5.2)$$

wobei δ_{ij} das Kronecker-Symbol⁵⁾ ist.

2. Einen klar definierten „Anfangszustand“ gibt es nicht. Die Kundenakquise beginnt mit Kunden, über die „mehr oder weniger“ bekannt ist. Das heißt, zum Beginn der Akquise können sich alle (potenziellen) Kunden grundsätzlich in jedem der nicht-beendeten Zustände befinden. Auf jeden Kunden wird dann eine dem Zustand, in dem er sich befindet, adäquate Akquiseaktion angewandt (deren Auswahl Gegenstand von Kapitel 6 ist).

Im Unterschied dazu beginnt eine Akquise mit dem einfachen oder mit dem erweiterten Akquiseplan stets mit dem Quellknoten (vgl. Def. 3.1), also einer ausgezeichneten Akquiseaktion. Die Merkmale der Kunden können dennoch gänzlich heterogen sein. Beim einfaches

5. siehe Fußnote 5 auf Seite 121

chen Akquiseplan zeigt dies die Notwendigkeit einer der Akquise vorangehenden Kundensegmentierung. Beim erweiterten Akquiseplan ist die Kundensegmentierung ein Bestandteil des Akquiseplans (siehe Abschnitt 5.2).

Bezüglich eines bestimmten Kunden endet die Akquise genau dann, wenn die Menge der Endzustände erreicht ist. Dann erzielt der Akquisiteur im günstigen Fall einen Ertrag, dessen Höhe vom jeweiligen Produkt bzw. von der konkreten Dienstleistung abhängt. Unter der Annahme, dass nur ein Produkttyp betrachtet wird, also z. B. nicht Versicherungen im Allgemeinen, sondern Lebensversicherungen, wird die genaue Höhe des Ertrags (z. B. eine Maklerprovision) von den Werten einiger Kunden-Merkmale abhängen, z. B. von der Höhe des Lohns oder Gehalts. Dieser Zusammenhang ist in der Regel nicht deterministisch. Man kann jedoch den Erwartungswert der Höhe des Ertrags als Funktion der positiven Endzustände angeben. Eine Möglichkeit zur Erhöhung der Genauigkeit dieser Schätzung ist die weitere Aufteilung des Körnchens⁶⁾ „posKE“, z. B. nach einzelnen Produkten bzw. Dienstleistungen oder Cluster davon, wie:

- Lebensversicherung (Premiumklasse): Endzustand „posKE mit hohem Ertrag“,
- Lebensversicherung (Standardklasse): Endzustand „posKE mit mittlerem Ertrag“,
- Lebensversicherung (Basisklasse): Endzustand „posKE mit geringem Ertrag“.

In folgender Definition wird ein „unscharfes Ziel“ definiert, also anstelle einer reellwertigen oder ganzzahligen Zielfunktion eine unscharfe Menge. Das hat folgende Gründe:

- Jede unscharfe Menge ist vermöge ihrer Zugehörigkeitsfunktion eine Zielfunktion über ihrer Grundmenge. Hier ist der Ertrag der Akquise abhängig von den Akquise-Zuständen in der Akquise-Zustandsmenge.
- Ein unscharfes Ziel kann sehr flexibel definiert werden. Neben der Definition über eine Ertragsfunktion (wie in der nachfolgenden Definition Def. 5.6) können weitere Bewertungskriterien in die Definition des unscharfen Ziels einfließen, z. B. schwer abzuschätzende künftige Erträge (aus Provisionen) oder eine wie auch immer geartete Gesamtbewertung des Kunden, z. B. des zukünftigen Kundenwertes („lifetime value“) oder subjektive Bewertungen.
- Die Verwendung eines unscharfen Ziels erlaubt die einfache Verwendung unscharfer Nebenbedingungen (Def. 5.7) und deren einheitliche Behandlung bei der Optimierung (mehr dazu in Abschnitt 5.4 und in Kapitel 6).

Definition 5.6: *Akquise-Ertragsfunktion, unscharfes Akquise-Ziel, unscharfes Akquise-Ziel zur Ertragsfunktion.*

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Akquise-Zustandsmenge $X = \{\sigma_1, \dots, \sigma_n\}$ und der Menge der positiven Akquise-Endzustände T gemäß Def. 5.2 und Def. 5.5. Sei $R \subseteq \mathbb{R}^+$ das abgeschlossene Intervall möglicher Ertragswerte⁷⁾. Dann ist eine Funktion $r : X \rightarrow R$ eine *Akquise-Ertragsfunktion*, wenn gilt:

6. englisch: granule; vgl. dazu Abschnitt 2.5, Seite 40

$$r(x) \begin{cases} \in \mathbb{R} & , x \in T_{\text{pos}} \\ = 0 & , \text{sonst} \end{cases}.$$

Ein *unscharfes Akquise-Ziel* G ist eine unscharfe Menge über der Zustandsmenge X , also $G \in \text{IF}(X)$ mit der Eigenschaft $\text{supp } G \subseteq T_{\text{pos}}$.

Das *unscharfe Akquise-Ziel* $G \in \text{IF}(X)$ zur *Ertragsfunktion* $r(x)$ wird durch Normalisierung wie folgt definiert:

$$\mu_G(x) := g\left[\frac{r(x)}{r_{\max}}\right] \quad (5.3)$$

mit $r_{\max} = \max R \neq 0$, und $g : [0, 1] \rightarrow [0, 1]$ ist monoton steigend mit $g(0) = 0$ und $g(1) = 1$ [11]. \square

Anmerkungen

1. Für $g \equiv 1$ entspricht das unscharfe Akquise-Ziel einer scharfen Ertragsfunktion, die jedem Endzustand den normierten Wert der Akquise-Ertragsfunktion r zuordnet.
2. Da die Akquise-Ertragsfunktion r nur über den Endzuständen definiert ist, kann sie die Kosten der Einzelentscheidungen im Akquise-Verlauf nicht berücksichtigen. Eine Gewinngröße als Differenz Ertrag – Kosten ist mit dem vorliegenden Ansatz nicht ohne weiteres direkt modellierbar. Im Falle der Akquise von Versicherungsprodukten kann der „Ertrag“ als normierte Provision verstanden werden.
3. Andererseits kann r auch eine Schätzung des langfristigen Kundenwerts, des so genannten „customer lifetime value“ umfassen, in den mögliche künftige Erträge eingehen.
4. Die Funktion g erlaubt es, das unscharfe Ziel $G \in \text{IF}(X)$ relativ flexibel aus der (über konkreten Geldwert) definierten Funktion r abzuleiten:
 - Für $g \equiv 1$ entspricht g einer Normierung der Akquise-Ertragsfunktion r (vgl. 1).
 - Wenn die Akquise-Ertragsfunktion r die Versicherungssumme bezeichnet und die Provision stufenweise festgelegt ist, so kann dies mit g modelliert werden:

Seien die Stufen der Versicherungssumme $[r_0, r_1]$ und $(r_{i-1}, r_i]$ für $i = 2, \dots, N$ mit $r_0 = 0$ und $r_N = r_{\max}$ und der Provisionsanteil jeweils $0 < \gamma_i < 1$ oder, als Geldwert: $\gamma_i r$ für $r \in (r_{i-1}, r_i]$. Der Geldwert der Provision ist somit 0 für $r = r_0 = 0$ und $\gamma_N r_{\max}$ für $r = r_{\max}$. Den Geldwert der Provision liefert die Funktion g :

7. Für die Betrachtungen hier ist es unerheblich, ob die Werte in \mathbb{R} mit einer Währungseinheit behaftet sind oder einheitenlose Vergleichswerte darstellen.

$$g(y) = \begin{cases} \gamma_N y & \text{für } y \in (r_{N-1}, r_N] \\ \gamma_{N-1} y & \text{für } y \in (r_{N-2}, r_{N-1}] \\ \vdots & \vdots \\ \gamma_1 y & \text{für } y \in [r_0, r_1] \end{cases},$$

so dass das unscharfe Akquise-Ziel als normalisierter Geldwert der Provision formuliert werden kann:

$$\mu_G(x) = g\left[\frac{r(x)}{r_{\max}}\right].$$

Alle Akquiseaktionen sind Beschränkungen (Nebenbedingungen) unterworfen, deren Ursachen Kosten oder Kunden-Eigenschaften und -Vorlieben sind oder zusätzliches Marketing-Wissen, das sich nicht in den Kunden-Eigenschaften widerspiegelt. So sollen Kosten möglichst gering sein, um Erträge nicht zu sehr zu schmälern und die Eigenschaften und Vorlieben der Kunden sind zu berücksichtigen, damit der Kunde schließlich die Menge positiver Endzustände erreicht. Da Kosten, Eigenschaften und Vorlieben bei der Neukundengewinnung graduelle Größen sind, werden sie durch unscharfe Mengen dargestellt.

Definition 5.7: *Unscharfe Akquise-Nebenbedingung.*

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Akquise-Zustandsmenge $X = \{\sigma_1, \dots, \sigma_n\}$ gemäß Def. 5.2, der Menge der nicht-beendeten Akquise-Zustände $\bar{T} = X \setminus T$ gemäß Def. 5.5 und der Akquise-Entscheidungsmenge $U = \{\alpha_1, \dots, \alpha_m\}$ gemäß Def. 5.3. Dann sind die *unscharfen Akquise-Nebenbedingungen* zustandsabhängige unscharfe Mengen über der Akquise-Entscheidungsmenge, $C(x) \in \mathcal{IF}(U)$, $x \in \bar{T}$. Ihre Zugehörigkeitsfunktionen werden mit $\mu_C(u | x)$ für gegebenes $x \in \bar{T}$ und alle $u \in U$ bezeichnet. \square

Anmerkung

Unscharfe Nebenbedingungen erlauben es, zu jedem nicht beendeten Zustand $x \in \bar{T}$ alle Akquiseaktionen $u \in U$ in eine von drei Kategorien einzuordnen:

- Die Akquiseaktion ist unzulässig, das heißt $\mu_C(u | x) = 0$.
- Die Akquiseaktion ist „unbeschränkt“ zulässig, das heißt $\mu_C(u | x) = 1$.
- Die Akquiseaktion ist „eingeschränkt zulässig“ bzw. „bedingt zulässig“, das heißt

$$0 < \mu_C(u | x) < 1.$$

Beispiel 5.4: In Abb. 5.6 sind zwei Beispiele für unscharfe Nebenbedingungen gezeigt. Zur Verdeutlichung wird angenommen, dass sie zu einer Neukundengewinnung für Lebensversicherungen gehören. Die Entscheidungsmenge U besitzt keine natürliche Ordnungsrelation. Die Entscheidungen haben nominales Skalenniveau⁸⁾. Es ist jedoch möglich, eine Ordnungsre-

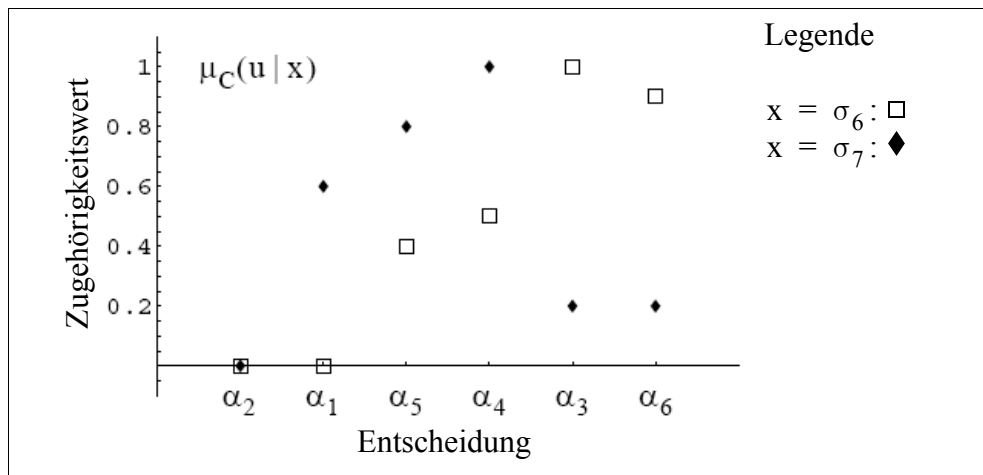


Abbildung 5.6: Unscharfe Nebenbedingungen bei der Neukundengewinnung für Lebensversicherungen.

	ξ_1 (Einkommen)	ξ_2 (Alter)	ξ_3 (Bildung)	ξ_4 (Interesse)
σ_6	hoch	mittleren Alters	Facharbeiterabschluss	unentschlossen
σ_7	gering	jung	Universitätsabschluss	hoch

Tabelle 5.2: Einige Zustände in einem Zustandsraum.

	Entscheidung	Kosten ^a
α_1	unpersönlicher Massenbrief (Mailing)	1,0
α_2	Telefonanruf	0,5
α_3	Brief mit Prospekt (Hochglanzpapier, keine Produktdetails)	6,0
α_4	Brief mit ausführlichen Produktinformationen	3,0
α_5	Brief mit konkretem Angebot für den Kunden	2,0
α_6	Besuch beim Kunden	20,0

Tabelle 5.3: Entscheidungen $\alpha_i \in U$, $i = 1, \dots, 6$ aus Abb. 5.6 und ihre Kosten.

a. einheitenlose relative Kosten

lation bezüglich einer Eigenschaft der Entscheidung einzuführen, z. B. nach dem Zeitaufwand der Aktion oder ihren Kosten. Letztere wurden in diesem Beispiel zur Ordnung der Entscheidungen herangezogen. Einzelheiten zu den Zuständen und Entscheidungen sind in Tab. 5.2 und in Tab. 5.3 angegeben.

Eine Regel, die jedem nicht-beendeten Akquise-Zustand unabhängig von der seit Akquisebeginn verstrichenen Zeit eine Akquise-Entscheidung zuordnet, wird im Folgenden definiert:

Definition 5.8: *Stationäre Akquise-Entscheidungsregel, stationäre Politik.*

Betrachtet werde eine Akquise von Kunden mit einer gegebenen Akquise-Zustandsmenge $X = \{\sigma_1, \dots, \sigma_n\}$ gemäß Def. 5.2, der Menge der nicht-beendeten Akquise-Zustände $\bar{T} = X \setminus T$ gemäß Def. 5.5 und der Akquise-Entscheidungs-
menge $U = \{\alpha_1, \dots, \alpha_m\}$ gemäß Def. 5.3. Dann ist eine *stationäre Akquise-Entscheidungsregel* $\pi : \bar{T} \rightarrow U$ eine Abbildung, die jedem nicht-beendeten Akquise-Zustand $x \in \bar{T}$ eine Akquise-Entscheidung $u \in U$ zuweist, das heißt $u = \pi(x)$. Sie wird auch als *stationäre Politik* bezeichnet. \square

Anmerkungen

1. Eine stationäre Politik hängt nur vom Zustand $x_t \in \bar{T}$ ab, nicht von der Zeit t ⁹⁾.
2. Die Definition kann auf alle Akquise-Zustände $x \in X$ erweitert werden, indem für $x \in T$ $\pi(x) := \alpha$ für ein beliebiges $\alpha \in U$ gesetzt wird, da gemäß (5.2) $p(x | x, u) = 1$ für alle $x \in T$ und alle $u \in U$.
3. Die Optimierung eines dynamischen Akquiseplans führt zu einer optimalen Politik. Der Begriff wird in Abschnitt 5.4 und in Kapitel 6 genau definiert.
4. Beim Entwurf eines gewöhnlichen Akquiseplans aus Abschnitt 3.2 kennt der Akquisiteur bei der Wahl einer Akquiseaktion alle möglichen „Vorgeschichten“ der Aktionen, das heißt alle Knotenfolgen, die die betrachtete Aktion enthalten. Der Akquisiteur hat sozusagen eine „globale Sichtweise“. Dagegen hat eine stationäre Politik nur eine auf einen Zustand beschränkte „lokale Sicht“ auf die Akquisition. In der Regel wird diese Sicht nicht ausreichen. Angenommen, vom Ausgangszustand bis zu dem Zustand, für den eine Aktion festzulegen ist, gibt es zwei Trajektorien, eine mit einem Zustandsübergang und eine mit drei Zustandsübergängen. Eine stationäre Politik würde dann im betrachteten Zustand für einen Kunden, der zuvor einmal mit einer Aktion bearbeitet worden ist, die gleiche Folgeaktion

9. [134] teilt Politiken in (a) Markowsche und (b) historische ein. Markowsche Politiken sind Entscheidungsregeln, die von vergangenen Zuständen und Aktionen nur durch den aktuellen Zustand abhängen. Historische Politiken sind Entscheidungsregeln, die von den vergangenen Zuständen und Aktionen abhängen. Beide unterteilt er nochmals in deterministische und stochastische. Bei letzter Kategorie wird durch die Politik keine Aktion bestimmt, sondern eine Wahrscheinlichkeitsverteilung für mögliche Aktionen. Eine stationäre Politik gemäß Def. 5.8 ist deterministisch. Im Vergleich mit Markowschen und historischen, deterministischen bzw. stochastischen Politiken ist die hier definierte stationäre Politik die spezifischste Entscheidungsregel, das heißt die Menge der stationären Politiken ist in den Mengen aller anderer Entscheidungsregeln enthalten.

auswählen wie für einen Kunden, der bereits dreimal mit einer Aktion bearbeitet worden ist. Das kann zu Fehlern bei der Akquise führen.

Eine Lösung des Problems ist die Einführung zusätzlicher Merkmalsvariablen mit Bezug zur Vorgeschichte. Damit der Zustandsraum nicht unverhältnismäßig stark vergrößert werden muss, ist es ungeeignet, die Vorgeschichte z. B. als Folge in einer Merkmalsvariablen abzulegen. Zweckmäßiger sind Merkmalsvariablen, die die Vorgeschichte zusammenfassen, was auch die Interpretation der Zustände erleichtert. In [130, 131] werden zu diesem Zweck

- die Neuigkeit einer Kundenreaktion und
- die Häufigkeit einer Kundenreaktion

als Merkmalsvariable festgelegt. Dies kann für die Kundengewinnung mit erweiterten Akquiseplänen übernommen werden:

- Die Merkmalsmenge der Neuigkeit einer Kundenreaktion ist eine Partition der Menge $\{0, 1, 2, \dots, N_{\max}\}$, wobei die Neuigkeit als die Anzahl der Akquiseaktionen ohne Kundenreaktion seit der letzten Kundenreaktion definiert wird. Dabei ist die Obergrenze N_{\max} ein vom Akquisiteur festzulegender Erfahrungswert.
- Die Merkmalsmenge der Häufigkeit einer Kundenreaktion ist eine Partition des Einheitsintervalls $[0, 1]$, wobei die Häufigkeit als

$$\frac{\text{Anzahl der Kundenreaktionen}}{\text{Anzahl der Aktionen}} \quad (5.4)$$

definiert wird. Alternativ kann die Häufigkeit einer Kundenreaktion auch als Anzahl der Kundenreaktionen definiert werden (wie in [62]). Auf die Bedeutung dieser Wahl wird im Abschnitt 6.5.3 eingegangen.

Die Merkmalsmengen der Neuigkeit und der Häufigkeit können auch grobkörniger partitioniert werden, z. B. durch Intervalle.

Beispiel 5.5: Die Tabelle 5.4 zeigt die Interpretation der Merkmale *Neuigkeit* und *Häufigkeit*. Der Vergleich von Kundensegment 1 und 2 zeigt, dass die Kunden des Kundensegments 2 trotz gleichen Wertes des Merkmals *Neuigkeit* „unzuverlässiger“ als die Kunden des Kundensegments 1 sind. Der Vergleich von Kundensegment 2 und 3 zeigt, dass eine Antwort auf die nächste Aktion trotz gleichen Wertes des Merkmals *Häufigkeit* eher von Kunden des Kundensegments 2 als von Kunden des Kundensegments 3 zu erwarten ist. Das Kundenmerkmal *Geldwert* ist hier nur der Vollständigkeit halber angegeben. Es ist Bestandteil der Kundensegmente aber für das Verständnis der Kundenmerkmale *Neuigkeit* und *Häufigkeit* belanglos.

Jetzt sind alle Voraussetzungen beschaffen, um einen dynamischen Akquiseplan definieren zu können.

Definition 5.9: *Dynamischer Akquiseplan.*

Betrachtet werde der zeitdiskrete dynamische Prozess der Kundenakquise, gegeben durch eine Menge von Akquise-Zuständen, $X = \{\sigma_1, \dots, \sigma_n\}$ gemäß Def. 5.2, eine darin enthaltene Menge von beendeten Akquise-Zuständen, $T \subseteq X$ gemäß Def. 5.5, die Komplementmenge der nicht beendeten Akquise-Zustände

Kunden-segment	Anzahl der Aktionen	Kundenmerkmal			Interpretation
		Neuigkeit	Häufigkeit	Geldwert	
1	5	0	1	50 €	Der Kunde hat auf alle Aktionen reagiert.
2	5	0	0,6	120 €	Der Kunde hat auf die letzte Aktion reagiert. Er hat auf zwei der vier vorangegangenen Aktionen reagiert.
3	5	2	0,6	30 €	Der Kunde hat auf die beiden letzten Aktionen nicht reagiert, aber auf alle vorangegangenen Aktionen.

Tabelle 5.4: Beispiel für die Interpretation der Merkmale *Neuigkeit* und *Häufigkeit*.

$\bar{T} = X \setminus T$, eine Menge von Akquise-Entscheidungen, $U = \{\alpha_1, \dots, \alpha_m\}$ gemäß Def. 5.3, die stochastischen Übergangsmatrizen, $P = \{P(\alpha) \mid \alpha \in U\}$ mit den Akquise-Übergangswahrscheinlichkeiten

$$p(x_{t+1} \mid x_t, u_t) \begin{cases} \geq 0 & \text{wenn } x_t \in \bar{T} \\ = 0 & \text{wenn } x_t \in T, x_{t+1} \neq x_t \\ = 1 & \text{wenn } x_t = x_{t+1} \in T \end{cases}$$

für $u_t \in U$ und $t = 0, 1, \dots$ gemäß Def. 5.4, die unscharfen Akquise-Nebenbedingungen, $C(x) \in \mathcal{IF}(U)$, $x \in \bar{T}$ gemäß Def. 5.7, das unscharfe Akquiseziel $G \in \mathcal{IF}(X)$ gemäß Def. 5.6 und die stationäre Akquise-Entscheidungsregel $\pi : X \rightarrow U$ gemäß Def. 5.8. Das Tupel (X, T, U, P, C, G, π) wird als *dynamischer Akquiseplan* bezeichnet. \square

Anmerkungen

1. Grundsätzlich können alle Akquise-Entscheidungen eines dynamischen Akquiseplans auf jeden Zustand (also jedes Kundensegment) angewendet werden. Das ist ein wesentlicher Unterschied zum einfachen Akquiseplan und zum erweiterten Akquiseplan, bei denen die Akquiseaktionen mit ganz bestimmten Kunden-Reaktionen (und damit Kundensegmenten) verknüpft sind.
2. Die Akquise-Nebenbedingungen erlauben eine Bewertung der Anwendung einer Akquiseaktion auf einen Akquise-Zustand. Dies schränkt die grundsätzliche Anwendbarkeit aller Akquise-Entscheidungen auf die Akquise-Zustände ein.
3. Eine Akquise-Entscheidungsregel π wählt schließlich zu jedem Akquise-Zustand genau eine Akquiseaktion aus – genau wie beim einfachen und beim erweiterten Akquiseplan.

4. Ein dynamischer Akquiseplan mit fester Akquise-Entscheidungsregel könnte grundsätzlich auch als einfacher Akquiseplan modelliert werden, falls alle Kunden zum Akquisebeginn genau einem Zustand zugeordnet werden können.

5.4 Unscharfe stochastische Optimierung mit dynamischen Akquiseplänen

Ein dynamischer Akquiseplan integriert Kundensegmentierung und Akquise in ähnlicher Weise wie ein erweiterter Akquiseplan (siehe Abb. 5.7, vgl. Abb. 5.1 und Abb. 5.2). Zunächst werden Segmente durch Definition des Zustandsraumes festgelegt. Danach werden während der Durchführung der Akquise die Kunden den Segmenten zugeordnet.

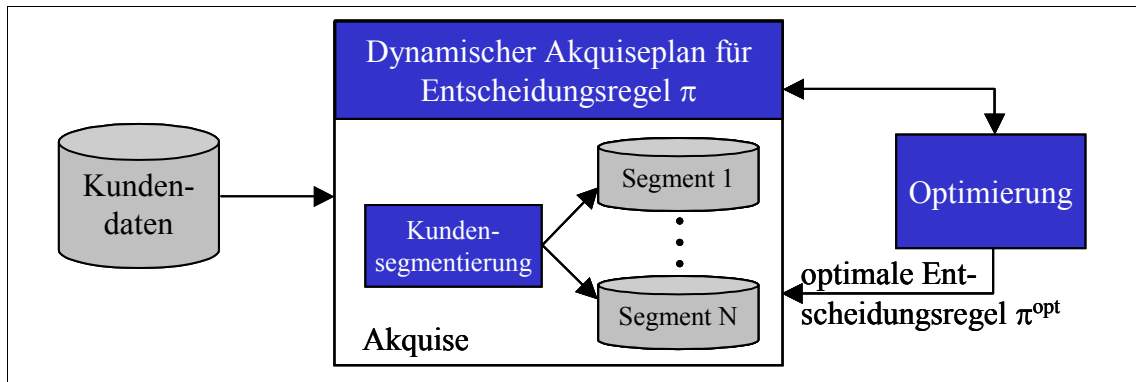


Abbildung 5.7: Dynamischer Akquiseplan mit in die Akquise integrierter Kundensegmentierung und darauf aufbauender Optimierung.

Eine der interessantesten Fragen im Zusammenhang mit der Durchführung einer Akquise dürfte die nach den „optimalen“ Aktionen sein, also jenen Aktionen, die die Akquise zu maximalem Gewinn führen. Dies wird erreicht, wenn die Aktionen so gewählt werden, dass möglichst viele Kunden in einen positiven Akquise-Endzustand gelangen, mit einem möglichst hohen Abschluss. Das im vorigen Abschnitt eingeführte Modell des dynamischen Akquiseplans ermöglicht die Formulierung eines entsprechenden Optimierungsproblems, vor allem, weil es in Gestalt der Akquise-Entscheidungsregel π eine Größe besitzt – eine Abbildung –, deren Optimierung der erwünschten Wahl optimaler Aktionen entspricht.

Somit besitzt das Optimierungsproblem die allgemeine Formulierung:

Berechne die Akquise-Entscheidungsregel eines gegebenen dynamischen Akquiseplans so, dass das zu erwartende unscharfe Akquiseziel unter Beachtung der unscharfen Akquise-Nebenbedingungen maximal wird.

Abb. 5.7 illustriert das Optimierungsproblem. Gesucht wird ein Verfahren, das aufbauend auf einem dynamischen Akquiseplan eine optimale Entscheidungsregel berechnet. Obige Formulierung des Optimierungsproblems enthält verschiedene bislang ungeklärte Begriffe, z. B. das „zu erwartende Akquiseziel“, das offensichtlich mit dem stochastischen Charakter der Akquise-Zustandsübergänge zu tun hat. Diese Begriffe werden in Kapitel 6 geklärt, wo auch die Lösung des Problems hergeleitet wird.

6 Mehrstufige stochastische Entscheidungsprozesse in unscharfer Umgebung

Kapitel 3 führt in die modellbasierte Neukundengewinnung mit Akquiseplänen ein. Im Kapitel 5 werden dynamische Akquisepläne vorgestellt, die geeignet sind, das Problem der Wahl optimaler Akquiseaktionen zu beschreiben. In diesem Modell wird die Neukundengewinnung als mehrstufiger stochastischer Entscheidungsprozess in einer unscharfen Umgebung mit implizit gegebener Prozess-Endzeit betrachtet. Sie ordnet sich in die mathematische Theorie mehrstufiger Entscheidungsprozesse („multi-stage decision processes“) ein, die 1957 von Bellman erstmals vorgestellt wurde und von ihm die Bezeichnung dynamische Optimierung („dynamic programming“) erhielt [9]. Ihre Grundbegriffe werden im Abschnitt 6.1 eingeführt. Im darauf folgenden Abschnitt 6.2 wird der Begriff der unscharfen Entscheidung („fuzzy decision“) vorgestellt, mit dessen Hilfe das dynamische Optimierungsproblem für unscharfe Umgebungen („fuzzy environment“) formuliert werden kann. Zur Begründung dieses Ansatzes wird auf Kapitel 5 verwiesen. Auch mit dieser Erweiterung und der Vielzahl möglicher mehrstufiger Entscheidungsprozesse folgt die Berechnung einer optimalen Politik im Wesentlichen der von Bellman eingeführten Methodik, worauf im Abschnitt 6.3 eingegangen wird. Für die Anwendung in der modellbasierten Neukundengewinnung ist es erforderlich, die bekannten Ansätze für implizit vorgegebene Prozess-Endzeiten zu erweitern. Die dafür in [161, 163, 164] vorgestellten Ansätze werden im Abschnitt 6.4 vollständig ausgearbeitet und genau beschrieben. Darauf aufbauende Algorithmen zur Berechnung optimaler Politiken werden im Abschnitt 6.5 vorgestellt sowie Abschätzungen des Berechnungsaufwands. Mit einem Fallbeispiel aus der Praxis der modellbasierten Neukundengewinnung schließt der Abschnitt 6.6 das Kapitel ab.

6.1 Grundbegriffe der dynamischen Optimierung

Die folgenden grundlegenden Definitionen der dynamischen Optimierung orientieren sich an [10].¹⁾

Definition 6.1: *Dynamisches System, Zustandsvektor, Dimension.*

Ein *dynamisches System* (S, X) besteht aus der Zustandsmenge X und einer Berechnungsvorschrift S , die jedem Zeitpunkt t einen *Zustandsvektor* („state vector“)

$$x(t) = (x_1(t), \dots, x_N(t))^T \in X, N \in \mathbb{N}$$

zuweist. N ist die *Dimension* des dynamischen Systems. □

1. Die Bezeichnungen in der Literatur sind nicht ganz einheitlich. Anstelle von „Prozess“ wird häufig auch die Bezeichnung „dynamisches System“ verwendet, z. B. in [56, 124].

Anmerkungen

1. Die Berechnungsvorschrift S kann unterschiedlicher Art sein, und die Zeit t kann stetig oder diskret betrachtet werden. Beispielsweise kann der Verlauf von $x(t)$ für $t \geq 0$ durch eine Differentialgleichung gegeben sein.
2. Im Folgenden werden *mehrstufige* dynamische Systeme betrachtet, das sind dynamische Systeme mit mehreren zeitdiskreten Zeitpunkten t bzw. Stufen. Weiterhin wird von einer endlichen Menge von Zustandsvektoren $X = \{\sigma_1, \dots, \sigma_n\}$, $n \in \mathbb{N}$ mit endlicher Dimension N ausgegangen, das heißt die σ_i , $i = 1, \dots, n$ bestehen aus jeweils N Komponenten.
3. Die Berechnungsvorschrift S wird im Folgenden näher spezifiziert durch Angabe einer Zustandsübergangsfunktion Φ bzw. von stochastischen Matrizen P , die eine Wahrscheinlichkeitsverteilung über aufeinanderfolgenden Zuständen beschreiben.

Definition 6.2: *Mehrstufiger deterministischer/stochastischer Entscheidungsprozess, Zustand, Anfangszustand, Endzustand, Stufenzahl, Zustandsvariable, Zustandsmenge/-raum, Zustandsübergang, Entscheidungsvariable, zulässige Entscheidung, Entscheidungsmenge/-raum, Entscheidung, Zielfunktion, Übergangswahrscheinlichkeit.*

Ein *mehrstufiger deterministischer Entscheidungsprozess* (S, X, U, Φ, R) („deterministic multistage decision process“) ist ein dynamisches System (S, X) mit folgenden Eigenschaften: Die Zustandsvektoren $x(t) \in X$ werden zu diskreten Zeitpunkten $t = 0, 1, 2, \dots, N$, $N \in \mathbb{N}$ betrachtet und *Zustände* $x_0, x_1, x_2, \dots, x_N$ („states“) genannt²). Der Zustand x_0 wird als *Anfangszustand* bezeichnet, x_N als *Endzustand* und N bezeichnet die *Stufenzahl*. x_t bezeichnet die *Zustandsvariable*. Sie nimmt Werte aus der Menge $X = \{\sigma_1, \dots, \sigma_n\}$ an, die *Zustandsmenge* oder *Zustandsraum* („state space“) genannt wird. Jeder Zustand $x_t \in X$, $t = 0, 1, \dots$ geht vermöge eines *Zustandsübergangs* Φ („state transition“) und der Entscheidung $u_t \in U_t(x_t) \subseteq U$ in einen Folgezustand $x_{t+1} = \Phi(x_t, u_t)$ über. Die *Entscheidungsvariable* u_t („decision variable“) nimmt Werte aus der Menge $U_t(x_t)$ an, der Menge der *zulässigen Entscheidungen* zur Zeit t im Zustand x_t , die in der *Entscheidungsmenge* $U = \{\alpha_1, \dots, \alpha_m\}$ enthalten ist. Letztere wird auch *Entscheidungsraum* („decision space“) genannt und ihre Elemente *Entscheidungen*. Der Bewertung des Prozesses für eine zunächst beliebige Anzahl von Zuständen

2. Die Zustände x_t , $t = 0, 1, \dots, N$ sind nicht mit den Komponenten des Zustandsvektors $x_i(t)$, $i = 1, \dots, N$ zu verwechseln. N ist nicht mit der Dimension der Zustandsvektoren (vgl. Def. 6.1) zu verwechseln.

dient die als *Zielfunktion* („objective function“) bezeichnete reellwertige Funktion $R = R(x_0, u_0, x_1, u_1, \dots, x_N) \in \mathbb{R}$.

Ein *mehrstufiger stochastischer Entscheidungsprozess* (S, X, U, P, R) („stochastic multistage decision process“) ist ein dynamisches System (S, X) das gegenüber einem mehrstufigen deterministischen Entscheidungsprozess (S, X, U, Φ, R) die folgende Besonderheit hat: Der Zustandsübergang ist nicht deterministisch. Statt dessen bestimmen die stochastischen Matrizen³⁾ $P(\alpha)$, $\alpha \in U$ mit den Elementen $p_{ij}(\alpha)$, $i, j = 1, \dots, n$ die Wahrscheinlichkeit, mit der ein Zustand $x_t = \sigma_i$ mit der Entscheidung $u_t = \alpha$ zum Folgezustand $x_{t+1} = \sigma_j$ übergeht, das heißt $p_{ij}(\alpha) = p(\sigma_j | \sigma_i, \alpha)$. Die $p_{ij}(\alpha)$ werden *Übergangswahrscheinlichkeiten* („transition probabilities“) genannt. \square

Anmerkungen

1. Nachfolgend werden Entscheidungsprozesse mit endlicher Entscheidungsmenge $U = \{\alpha_1, \dots, \alpha_m\}$ betrachtet.
2. Die Zielfunktion bewertet nicht nur Endzustände, wie die Akquise-Ertragsfunktion gemäß Def. 5.6, sondern auch Zwischenzustände. Damit können in der Zielfunktion auch Kosten von Entscheidungen berücksichtigt werden, die beim dynamischen Akquiseplan in die unscharfen Akquise-Nebenbedingungen gemäß Def. 5.7 eingehen.

Definition 6.3: *Politik, stationäre Politik, optimale Politik, Menge der Politiken, Politikraum (für scharfe Entscheidungsprozesse).*

Sei (S, X, U, Φ, R) bzw. (S, X, U, P, R) ein mehrstufiger deterministischer bzw. stochastischer Entscheidungsprozess. Eine Folge von Funktionen

$$\pi = \{\pi_0, \pi_1, \dots, \pi_{N-1}\},$$

wird *Politik* („policy“) genannt, wenn durch $\pi_0: X \rightarrow U_0$ und $\pi_t: X \times U_0 \times \dots \times X \times U_{t-1} \times X \rightarrow U_t$ für $t = 1, \dots, N-1$ für die Zustände $x_0, \dots, x_t \in X$ und die zulässigen Entscheidungen $u_0, \dots, u_{t-1} \in U$ die Entscheidung $u_t \in U_t$ festgelegt wird, $t = 0, \dots, N-1$. Eine Politik π , die nur aus $\pi_t: X \rightarrow U_t$, $t = 0, \dots, N-1$ besteht und von t unabhängig ist, heißt *stationäre Politik*. Da die Funktionen π_t , $t = 0, \dots, N-1$ identisch sind, werden Politik π und Funktionen π_t gleichgesetzt, das heißt $\pi_t = \pi$ für alle $t = 0, 1, \dots, N-1$. Eine Politik, die die Zielfunktion R (deterministischer Fall) bzw. den Erwartungswert der Zielfunktion ER (stochastischer Fall) maximiert⁴⁾, heißt *optimale Politik*.

3. vgl. Fußnote 11 auf Seite 62

π^{opt} . Die *Menge der Politiken* („policy set“) eines Entscheidungsprozesses wird mit Π bezeichnet und auch *Politikraum* genannt („policy space“). \square

Anmerkung

Im Allgemeinen kann von optimaler Politik π^{opt} nur im Zusammenhang mit einem Anfangszustand $x_0 \in X$ gesprochen werden. Es kann aber auch optimale Politiken geben, die für beliebige Anfangszustände optimal sind. Diese stehen im Mittelpunkt der folgenden Betrachtungen.

Die Formulierung des Optimierungsproblems für deterministische bzw. stochastische mehrstufige Entscheidungsprozesse mit vorgegebener Stufenzahl bzw. implizit gegebener Prozess-Endzeit folgt im Abschnitt 6.3. Zum Abschluss der Grundbegriffe wird das folgende *Optimalitätsprinzip* eingeführt, dem in der dynamischen Optimierung große Bedeutung zukommt.

Optimalitätsprinzip („principle of optimality“): *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* [9]

Anmerkung

Das Optimalitätsprinzip ist nicht allgemeingültig, sondern vom konkreten Problem abhängig. Ein hinreichendes Kriterium gibt der folgende Satz aus [88]⁵⁾.

Satz 6.1: Sei $R : X^{N+1} \times U^N \rightarrow \mathbb{R}$ die Zielfunktion eines mehrstufigen deterministischen Entscheidungsprozesses mit der Zustandsmenge X , der Entscheidungsmenge U und fester Stufenzahl $N \in \mathbb{N}$ ⁶⁾ gemäß Def. 6.2 mit der Gestalt

$$\begin{aligned} R(x_0, u_0, x_1, u_1, \dots, x_{N-1}, u_{N-1}, x_N) \\ = \rho(r_0(x_0, u_0), r_2(x_1, u_1), \dots, r_{N-1}(x_{N-1}, u_{N-1}), r_N(x_N)) \end{aligned}$$

wobei $r_t : X \times U_t \rightarrow \mathbb{R}$ mit $U_t \subseteq U$ für $t = 0, \dots, N-1$ der Beitrag von Zustand x_t unter der Entscheidung u_t zur Zielfunktion R ist und $r_N(x_N)$ der Beitrag des Endzustandes. Dabei gehe jeder Zustand x_t vermöge der Zustandsübergangsfunktion Φ aus x_{t-1} und u_{t-1} , $t = 1, \dots, N$ hervor: $x_t = \Phi(x_{t-1}, u_{t-1})$. Wenn es zwei Funktionen φ_0 und ψ_1 gibt, so dass R trennbar ist, das heißt

-
4. In der Literatur wird die Problemstellung auch oft so formuliert, dass die Zielfunktion bzw. ihr Erwartungswert zu minimieren ist. Das zeigt nur die Abhängigkeit der Problemstellung von der Interpretation der Zielfunktion. Stellt sie einen Ertrag oder eine Belohnung („reward“) dar – also eine positive Bewertung wie in der vorliegenden Arbeit – dann ist sie zu maximieren (z. B. auch in [134]). Stellt sie eine „Kostenfunktion“ dar (wie z. B. in [124]), dann ist ihre Minimierung erwünscht.
 5. Die Unbedingtheit des Optimalitätsprinzips in obiger Formulierung hat ihre Ursache darin, dass alle in [9] besprochenen Problemen dieses Kriterium erfüllen [88].
 6. Das heißt, das dynamische System durchläuft ausgehend vom Anfangszustand x_0 die Zustände x_1, \dots, x_N .
-

$$\begin{aligned} & \varphi(r_0(x_0, u_0), r_2(x_1, u_1), \dots, r_{N-1}(x_{N-1}, u_{N-1}), r_N(x_N)) \\ &= \varphi_0(r_0(x_0, u_0), \psi_1(r_1(x_1, u_1), \dots, r_{N-1}(x_{N-1}, u_{N-1}), r_N(x_N))) \end{aligned}$$

und wenn φ_0 für jedes r_0 monoton wachsend in ψ_1 ist, dann gilt das *Optimalitätsprinzip*, das heißt

$$\begin{aligned} & \max_{u_0 \in U_0, \dots, u_{N-1} \in U_{N-1}} \varphi(r_0(x_0, u_0), r_1(x_1, u_1), \dots, r_{N-1}(x_{N-1}, u_{N-1}), r_N(x_N)) \\ &= \max_{u_0 \in U_0} [\varphi_0(r_0(x_0, u_0), \max_{u_1 \in U_1, \dots, u_{N-1} \in U_{N-1}} \psi_1(r_1(x_1, u_1), \dots, r_{N-1}(x_{N-1}, u_{N-1}), r_N(x_N)))] \end{aligned}$$

□

6.2 Dynamische Optimierung in unscharfer Umgebung

Die Erweiterung der dynamischen Optimierung auf Probleme der Entscheidungsfindung bei Unschärfe wurde von Bellman und Zadeh mit dem Aufsatz „Decision-making in a fuzzy environment“ begonnen [11]⁷⁾. Als Motivation führen sie die Tatsache an, dass bei den meisten echten Problemen der Entscheidungsfindung Ziele, Nebenbedingungen und Folgen möglicher Aktionen nicht genau bekannt sind. Eine Ursache dafür ist das Vorhandensein von Unschärfe (vgl. Abschnitt 1.1), die mit der Wahrscheinlichkeitstheorie nicht abgebildet werden kann (vgl. Abschnitt 2.4). Deshalb führen sie den Begriff der *unscharfen Umgebung* („fuzzy environment“) ein, die aus *unscharfen Zielen* („fuzzy goals“), *unscharfen Nebenbedingungen* („fuzzy constraints“), und *unscharfen Entscheidungen* („fuzzy decision“) besteht [86]. „Unscharfe Umgebung“ bedeutet nicht, dass der betrachtete Entscheidungsprozess unscharf ist. Er kann deterministisch, stochastisch (wie in Def. 6.2 definiert) oder auch unscharf sein.

Die Definitionen der folgenden grundlegenden Begriffe folgen der Darstellung in der Originalarbeit [11] und den Monographien [84, 86].

Definition 6.4: *Unscharfes Ziel.*

Ein *unscharfes Ziel* ist eine unscharfe Menge G über einer Menge von „Wahlmöglichkeiten“ X , das heißt $G \in \text{IF}(X)$. □

Definition 6.5: *Unscharfe Nebenbedingung.*

Eine *unscharfe Nebenbedingung* ist eine unscharfe Menge C über einer Menge von „Wahlmöglichkeiten“ X , das heißt $C \in \text{IF}(X)$. □

7. Bereits 1969 wurde in [31] die wahrscheinlich erste Erweiterung der dynamischen Optimierung für die Berücksichtigung von Unschärfe vorgestellt. Die meisten Arbeiten der letzten 30 Jahre im Bereich der unscharfen dynamischen Optimierung beziehen sich aber auf den anderen Ansatz von Bellman und Zadeh, der auch der vorliegenden Arbeit zu Grunde liegt.

Anmerkungen

1. Die Definitionen sind identisch. Zwischen unscharfem Ziel und unscharfen Nebenbedingungen besteht kein grundsätzlicher Unterschied.
2. Die Zugehörigkeitswerte $\mu_G(x)$ bzw. $\mu_C(x)$ geben an, „zu welchem Grad das Ziel bzw. die Nebenbedingung durch $x \in X$ zufrieden gestellt wird“.
3. Ein unscharfes Ziel G kann aus einer scharfen Zielfunktion R gewonnen werden. Angenommen $R : X \rightarrow \mathbb{R}_0^+$ ordnet jeder Auswahl $x \in X$ eine positive reelle Zahl $R(x) \in \mathbb{R}$ zu und R ist beschränkt, dann kann – vorausgesetzt $\max R(x) \geq 0$ – ein unscharfes Ziel durch Normalisierung von R gewonnen werden:

$$\mu_G(x) = \frac{R(x)}{\max_{y \in X} R(y)} . \quad (6.1)$$

4. Eine unscharfe Nebenbedingung $C \in \text{IF}(X)$ kann als Verallgemeinerung der Menge zulässiger Punkte aus X , ihre Zugehörigkeitsfunktion μ_C als charakteristische Funktion der Menge zulässiger Punkte und die Punkte in $X \setminus \text{supp} C$ können als nicht zulässig betrachtet werden.
5. Unscharfe Ziele und unscharfe Nebenbedingungen werden ähnlich behandelt. Das vereinfacht die Definition der zugehörigen unscharfen Entscheidung.

Definition 6.6: *Unscharfe Entscheidung, maximierende Entscheidung.*

Gegeben seien n unscharfe Ziele G_1, \dots, G_n und m unscharfe Nebenbedingungen C_1, \dots, C_m über ein und derselben Menge von „Wahlmöglichkeiten“ X . Dann ist die zugehörige *unscharfe Entscheidung* die unscharfe Menge $D \in \text{IF}(X)$ mit der Zugehörigkeitsfunktion

$$\mu_D(x) = \mu_{G_1}(x) * \dots * \mu_{G_n}(x) * \mu_{C_1}(x) * \dots * \mu_{C_m}(x) \quad (6.2)$$

für jedes $x \in X$, wobei $*$ ein „Aggregationsoperator“ ist⁸⁾ (siehe Anmerkungen unten).

Eine *maximierende Entscheidung* („maximizing decision“) ist ein $x \in X$ für das die unscharfe Entscheidung $\mu_D(x)$ bzw., falls Zufallsgrößen vorhanden sind, der Erwartungswert der unscharfen Entscheidung $E\mu_D(x)$ maximal wird. Sie wird mit x^{opt} bezeichnet⁹⁾. □

8. Ganz allgemein beschreiben Bellman und Zadeh eine Entscheidung als das Zusammenkommen von Zielen und Nebenbedingungen: „Decision = Confluence of Goals and Constraints“ [11].

9. Eine maximierende Entscheidung muss nicht eindeutig sein; es kann mehrere maximierende Entscheidungen geben.

Beispiel 6.1: Abb. 6.1 zeigt eine unscharfe Entscheidung im Falle eines unscharfen Zieles und einer unscharfen Nebenbedingung ($n = m = 1$). x^{opt} bezeichnet die maximierende Entscheidung.

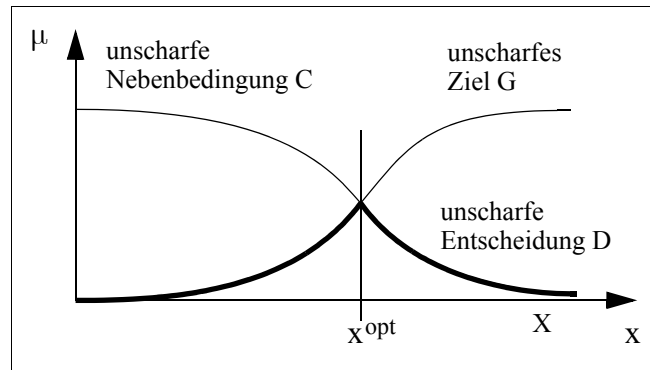


Abbildung 6.1: Unscharfe Entscheidung = „Zusammenkommen“ von unscharfem Ziel und unscharfer Nebenbedingung mit der Minimumbildung als Aggregationsoperator. x^{opt} bezeichnet die maximierende Entscheidung⁸⁾.

Anmerkungen zu Def. 6.6

1. Bellman und Zadeh gebrauchen einen einheitlichen Aggregationsoperators für die unscharfen Ziele $\mu_{G_1}(x) * \dots * \mu_{G_n}(x)$ einerseits, die unscharfen Nebenbedingungen $\mu_{C_1}(x) * \dots * \mu_{C_m}(x)$ andererseits und auch für ihre Verknüpfung zu einer unscharfen Entscheidung $\mu_D(x) = \mu_{G_1}(x) * \dots * \mu_{G_n}(x) * \mu_{C_1}(x) * \dots * \mu_{C_m}(x)$ und schlagen dafür Operatoren vor, die einem logischen „und“ bzw. dem Durchschnitt der betrachteten unscharfen Mengen entsprechen (vgl. Abschnitt 2.1). Das sind vor allem t-Normen. Kacprzyk spricht von einer unscharfen Entscheidung vom t-Norm-Typ („t-norm type fuzzy decision“) [86]. Diese Festlegung wird dem konkreten Problem jedoch nicht immer gerecht. In [194] werden Beispiele angeführt, die den Gebrauch so genannter *kompensatorischer unscharfer Operatoren*¹⁰⁾ nahelegen. Als Verallgemeinerung von Def. 6.6 wird dort folgende Interpretation für das „Zusammenkommen von Zielen und Nebenbedingungen“ vorgeschlagen.

Definition 6.7: *Unscharfe Entscheidung* (Verallgemeinerung von Def. 6.6).

Gegeben seien n unscharfe Ziele G_1, \dots, G_n und m unscharfe Nebenbedingungen C_1, \dots, C_m über einer Menge von Wahlmöglichkeiten X . Dann ist die zugehörige *unscharfe Entscheidung* die unscharfe Menge $D \in \text{IF}(X)$ mit der Zugehörigkeitsfunktion

10. Das sind Operatoren, die in besonderer Weise auf konkrete Situationen im Alltag passen und bei denen strukturalgebraische Eigenschaften eine untergeordnete Rolle spielen, z. B. der min-max-Kompensationsoperator $K_\gamma(x, y) = [\min(x, y)]^{1-\gamma} \cdot [\max(x, y)]^\gamma$, $\gamma \in [0, 1]$. Kleinere γ -Werte präferieren den Minimum-Operator im Sinne des logischen „und“, größere γ -Werte präferieren den Maximum-Operator im Sinne des logischen „oder“ [25].

$$\mu_D(x) = [\diamond_i \mu_{G_i}(x)] \bullet [\blacksquare_j \mu_{C_j}(x)] \quad (6.3)$$

für jedes $x \in X$, wobei \bullet , \diamond_i , \blacksquare_j geeignete, gegebenenfalls kontextabhängige „Aggregatoren“ bezeichnen [194]. \square

2. Wird als Aggregationsoperator in (6.2) eine t-Norm gewählt, dann ist

$$\mu_D(x) = 0 \text{ für alle } x \in X \setminus \bigcap_{j=1}^m \text{supp } C_j,$$

das heißt alle Punkte aus X , die nicht in den Trägern aller unscharfen Nebenbedingungen liegen, sind nicht zulässig (Zugehörigkeitswert Null). Diese scharfe Abgrenzung der Menge zulässiger Punkte kann durch die Wahl anderer Aggregationsoperatoren aufgeweicht werden, wie es insbesondere in (6.3) möglich ist. Werden dort \bullet und \blacksquare_j geeignet gewählt, kann z. B. die Verletzung von

$$x \in \bigcap_{j=1}^m \text{supp } C_j$$

durch einen höheren Zugehörigkeitswert bei den unscharfen Zielen kompensiert werden.

3. Wird als Aggregationsoperator in (6.2) die Minimum-t-Norm gewählt, dann ist der Grad der Befriedigung aller Nebenbedingungen, $\mu_{C_1}(x) * \dots * \mu_{C_m}(x)$ gleich dem Grad der am wenigsten befriedigten Nebenbedingung. Infolgedessen kann es vorkommen, dass Entscheidungen, von denen augenscheinlich eine besser als die andere ist, gleich bewertet werden. Dubois und Prade untersuchen das Problem als Bestandteil eines „calculus of flexible constraints“, vergleichen den Ansatz von Bellman und Zadeh mit einer *Pareto-Ordnung* und stellen als Alternativen die Relationen *Discrimin* und *Leximin* vor [39].
4. In Anwendungen sind häufig die unscharfen Ziele und die unscharfen Nebenbedingungen nicht über der gleichen Menge definiert. Zum Beispiel sind bei den nachfolgend betrachteten mehrstufigen Entscheidungsprozessen die unscharfen Ziele über der Zustandsmenge und die unscharfen Nebenbedingungen über der Entscheidungsmenge definiert.

Angenommen das unscharfe Ziel sei $G \in \text{IF}(X)$ und die unscharfe Nebenbedingung $C \in \text{IF}(U)$ und eine Funktion $f: X \rightarrow U$, $u = f(x)$ sei bekannt. Dann kann über $\mu_{C'}(x) = \mu_C(f(x))$ eine unscharfe Nebenbedingung $C' \in \text{IF}(X)$ definiert werden, so dass die Definitionen 6.6 oder 6.7 anwendbar sind. Diese Rolle dieser Funktion wird im Folgenden von einer Politik $\pi_t: X \rightarrow U$ übernommen, die wie folgt definiert ist.

Definition 6.8: *Politik, stationäre Politik, optimale Politik (für unscharfe Entscheidungsprozesse).*

Gegeben sei ein mehrstufiger deterministischer oder stochastischer Entscheidungsprozess in unscharfer Umgebung mit unscharfen Nebenbedingungen über U (Def. 6.5) und unscharfen Zielen über X (Def. 6.4). Entsprechend den konkreten Eigenschaften (siehe unten) des Entscheidungsprozesses sei eine unscharfe Entscheidung (Def. 6.6, Def. 6.7) definiert. Eine Folge von Funktionen

$$\pi = \{ \pi_0, \pi_1, \dots, \pi_{N-1} \},$$

wird *Politik* („policy“) genannt, wenn durch $\pi_0 : X \rightarrow U$ und $\pi_t : X^t \times U^{t-1} \rightarrow U$ für $t = 1, \dots, N-1$ für $x_0, \dots, x_t \in X$ und $u_0, \dots, u_{t-1} \in U$ der Wert der Entscheidungsvariablen $u_t \in U$ festgelegt wird, $t = 0, \dots, N-1$. Eine Politik π , die nur aus $\pi_t : X \rightarrow U$, $t = 0, \dots, N-1$ besteht und von t unabhängig ist, heißt *stationäre Politik*. Da die Funktionen π_t , $t = 0, \dots, N-1$ einer stationären Politik identisch sind, werden Politik π und Funktionen π_t gleichgesetzt, das heißt $\pi_t = \pi$ für alle $t = 0, 1, \dots, N-1$. Eine Politik, die eine maximierende Entscheidung (Def. 6.6) bezüglich eines Anfangszustands $x_0 \in X$ berechnet heißt *optimale Politik* π^{opt} bezüglich x_0 . Eine Politik, die eine maximierende Entscheidung für beliebige Anfangszustände berechnet heißt *optimale Politik*. \square

Diese Definition ist bewusst sehr allgemein gehalten, ohne Angabe einer bestimmten unscharfen Entscheidung, weil es eine Vielzahl mehrstufiger Entscheidungsprozesse in einer unscharfen Umgebung gibt, die sich darin unterscheiden. Sie werden üblicherweise nach folgenden Gesichtspunkten klassifiziert [84, 85, 86]:

a) Festlegung der Prozess-Endzeit:

- fest vorgegeben, das heißt der Prozess ist nach einer vorgegebenen Anzahl von Zustandsübergängen beendet;
- implizit vorgegeben, das heißt der Prozess endet, wenn der Prozesszustand eine Menge von Endzuständen („termination set“) bzw. einen bestimmten Endzustand („termination state“) erreicht hat;
- unscharfe Prozess-Endzeit, das heißt die Zeit nach der der Prozess endet ist eine unscharfe Menge;
- unendlich, das heißt der Prozess endet nicht.

b) Art des Prozesses:

- deterministisch, das heißt es gibt eine Zustandsübergangsfunktion Φ ;
- stochastisch, das heißt es gibt eine oder mehrere stochastische Matrizen mit Übergangswahrscheinlichkeiten;
- unscharfer Prozess, das heißt die Zustände des Prozesses sind unscharfe Mengen.

c) Art des Aggregationsoperators: t-Norm, s-Norm, kompensatorische Operatoren¹⁰⁾ u. a.

Als weiterer Klassifikationsgesichtspunkt kann die Definition der unscharfen Ziele und Nebenbedingungen dienen.

a) Unscharfe Ziele $G \in \text{IF}(X)$ können

- für jeden Prozess-Schritt vereinbart werden, das heißt $G = G(t)$, $t = 0, 1, \dots$,
- für jeden Prozess-Zustand, das heißt $G = G(x_t)$, $x_t \in X$,

- oder ggf. nur für den Prozess-Endzustand;

b) unscharfe Nebenbedingungen $C \in \mathbb{IF}(U)$ können

- für jeden Prozess-Schritt vereinbart werden, das heißt $C = C(t)$, $t = 0, 1, \dots$,
- oder für jeden Prozess-Zustand, das heißt $C = C(x_t)$, $x_t \in X$.

Entsprechend der Vielzahl an Wahlmöglichkeiten gibt es zahlreiche unterschiedliche Definitionen des konkreten dynamischen Optimierungsproblems. Ein Überblick würde den Rahmen dieser Arbeit sprengen und kann z. B. in [84] nachgelesen werden. Trotz der Vielfalt der Probleme haben die Lösungsansätze einige Gemeinsamkeiten, die auch der Erweiterung für mehrstufige stochastische Entscheidungsprozesse in unscharfer Umgebung und implizit vorgegebener Prozess-Endzeit im Abschnitt 6.4 zu Grunde liegen. Auf diese Gemeinsamkeiten geht der nächste Abschnitt ein.

6.3 Methodik der dynamischen Optimierung mehrstufiger Entscheidungsprozesse mit fester Stufenzahl

Zu den Grundprinzipien der dynamischen Optimierung mehrstufiger Entscheidungsprozesse gehört das Rekursionsprinzip. Es beinhaltet die Zurückführung eines Problems mit N Stufen auf ein solches mit $N - 1$ Stufen [20]. Zur Veranschaulichung werden die folgenden N -stufigen Entscheidungsprozesse betrachtet:

- Abschnitt 6.3.1: Deterministischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl, (S, X, U, Φ, R) ;
- Abschnitt 6.3.2: Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl, (S, X, U, P, R) ;
- Abschnitt 6.3.3: Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl in unscharfer Umgebung.

An diesen Beispielen sollen zum einen die Analogien zwischen deterministischem und stochastischem Entscheidungsprozess verdeutlicht werden und zum anderen die Analogien zwischen Entscheidungsprozessen mit unscharfer Umgebung und solchen mit scharfer Umgebung.

Das Rekursionsprinzip entfaltet seine Stärke durch wiederholte Anwendung des *Optimalitätsprinzip* (siehe Seite 242).

6.3.1 Deterministischer Entscheidungsprozess mit fester Stufenzahl

Sei (S, X, U, Φ, R) ein deterministischer mehrstufiger Entscheidungsprozess mit $N \in \mathbb{IN}$ Stufen, das heißt insbesondere, dass $x_{t+1} = \Phi(x_t, u_t)$ für $t = 0, \dots, N - 1$. Sei R eine additive Zielfunktion,

$$R(x_0, u_0, \dots, x_{N-1}, u_{N-1}, x_N) = \sum_{t=0}^{N-1} r_t(x_t, u_t) + r_N(x_N) \quad (6.4)$$

und sei

$$R_{N-t}(x_t, u_t, \dots, x_{N-1}, u_{N-1}, x_N) := \sum_{\tau=t}^{N-1} r_\tau(x_\tau, u_\tau) + r_N(x_N) \quad (6.5)$$

der Anteil an der Zielfunktion ab der Stufe t für $0 \leq t \leq N$. Damit ist $R_0 = r_N$ und $R_N = R$.

Bezeichne $R_{N-t}^{\max}(x_t)$ den maximalen Wert von (6.5), also

$$R_{N-t}^{\max}(x_t) := \max_{u_t, \dots, u_{N-1} \in U} R_{N-t}(x_t, u_t, \dots, x_{N-1}, u_{N-1}, x_N), \quad t = 0, \dots, N-1 \quad (6.6)$$

und

$$R_0^{\max}(x_N) = R_0(x_N) = r_N(x_N). \quad (6.7)$$

Unter Verwendung dieser Schreibweise kann das Maximum der Zielfunktion für einen Anfangszustand $x_0 \in X$ mit Hilfe des folgenden Satzes berechnet werden.

Satz 6.2: Für (S, X, U, Φ, R) mit R wie in (6.4) und $R_{N-t}^{\max}(x_t)$ wie in (6.6) und (6.7) gilt

$$R_{N-t}^{\max}(x_t) = \max_{u_t \in U} [r_t(x_t, u_t) + R_{N-(t+1)}^{\max}(\Phi(x_t, u_t))], \quad t = 0, \dots, N-1 \quad (6.8)$$

und

$$R_0^{\max}(x_N) = r_N(x_N). \quad (6.9)$$

□

Beweis: Der Beweis wird direkt geführt:

Gleichung 6.9 folgt unmittelbar aus (6.7).

Nach (6.6) ist mit (6.5) für $t = 0, \dots, N-1$

$$R_{N-t}^{\max}(x_t) = \max_{u_t, \dots, u_{N-1} \in U} [r_t(x_t, u_t) + \dots + r_{N-1}(x_{N-1}, u_{N-1}) + r_N(x_N)],$$

woraus wegen der Assoziativität der Maximumbildung bezüglich der Addition

$$\begin{aligned} R_{N-t}^{\max}(x_t) &= \max_{u_t \in U} \left[r_t(x_t, u_t) \right. \\ &\quad \left. + \max_{u_{t+1}, \dots, u_{N-1} \in U} [r_{t+1}(x_{t+1}, u_{t+1}) + \dots + r_{N-1}(x_{N-1}, u_{N-1}) + r_N(x_N)] \right] \end{aligned}$$

folgt. Mit (6.6), (6.5) und $x_{t+1} = \Phi(x_t, u_t)$ ergibt sich schließlich

$$R_{N-t}^{\max}(x_t) = \max_{u_t \in U} [r_t(x_t, u_t) + R_{N-(t+1)}^{\max}(\Phi(x_t, u_t))] .$$

■

Anmerkungen zu Satz 6.2

1. Mit Hilfe des Satzes kann rekursiv das Maximum der Zielfunktion R für jeden Anfangszustand $x_0 \in X$ berechnet werden:

- Die Berechnung beginnt mit (6.9). Für jeden Zustand aus $X = \{\sigma_1, \dots, \sigma_n\}$ wird

$R_0^{\max}(x_N) = r_N(x_N)$ berechnet. Das Ergebnis ist

$$R_0^{\max}(\sigma_1) = r_N(\sigma_1), \dots, R_0^{\max}(\sigma_n) = r_N(\sigma_n). \quad (6.10)$$

- Im nächsten Schritt werden mit (6.8) für jeden Zustand aus $X = \{\sigma_1, \dots, \sigma_n\}$ die Werte $R_1^{\max}(\sigma_1), \dots, R_1^{\max}(\sigma_n)$ berechnet. Dazu ist für jeden Zustand der Wert der Entscheidungsvariablen u_1 aus $U = \{\alpha_1, \dots, \alpha_m\}$ zu bestimmen, z. B. für den Zustand σ_1 :

$$R_1^{\max}(\sigma_1) = \max_{u_1 \in U} [r_{N-1}(\sigma_1, u_1) + R_0^{\max}(\Phi(\sigma_1, u_1))].$$

Die möglichen Werte für $R_0^{\max}(\Phi(\sigma_1, u_1))$ sind die zuvor berechneten Werte (6.10).

- Diese Berechnung wiederholt sich in allen weiteren Schritten entsprechend (6.8), bis schließlich mit

$$R_N(\sigma_1), \dots, R_N(\sigma_n) \quad (6.11)$$

für alle möglichen Anfangszustände $x_0 \in X$ das Maximum der Zielfunktion R bekannt ist.

2. Neben dem Maximum der Zielfunktion kann mit Hilfe des Satzes auch eine optimale Politik für einen beliebig gegebenen Anfangszustand $x_0 \in X$ berechnet werden. Sie erfolgt in zwei Phasen, einer Rückwärtsphase und einer Vorwärtsphase.

- In der Rückwärtsphase wird in jedem Berechnungsschritt t aus 1. die für jeden Zustand $x_t \in X$ bestimmte optimale Entscheidung $u_t \in U$ festgehalten:

$$\pi_t^{\text{opt}}(x_t) := \arg \max_{u_t \in U} [r_t(x_t, u_t) + R_{N-(t+1)}^{\max}(\Phi(x_t, u_t))], \quad t = 0, \dots, N-1.$$

Das Ergebnis des letzten Schrittes ist

$$\pi_0^{\text{opt}}(\sigma_1), \dots, \pi_0^{\text{opt}}(\sigma_n).$$

- Nach Abschluss der Rückwärtsphase ist die optimale Politik $\pi^{\text{opt}} = (\pi_0^{\text{opt}}, \dots, \pi_{N-1}^{\text{opt}})$ bekannt. Ihre Werte für einen gegebenen Anfangszustand $x_0 \in X$ werden in der Vorwärtsphase bestimmt: Im ersten Schritt wird der Wert von π_0^{opt} für x_0 abgelesen: $u_0^{\text{opt}} = \pi_0^{\text{opt}}(x_0)$. Daraus wird der Folgezustand berechnet: $x_1 = \Phi(x_0, u_0)$ und für x_1 der Wert $u_1^{\text{opt}} = \pi_1^{\text{opt}}(x_1)$ abgelesen. Dieses Verfahren wird bis zum Schritt $t = N - 1$ fortgesetzt. Das Ergebnis ist:

$$\pi^{\text{opt}}(x_0) = (u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}).$$

Durch Einsetzen der $u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}$ und x_0, \dots, x_N in (6.4) wird der Maximalwert der Zielfunktion für den Anfangszustand x_0 (der in (6.11) enthalten ist) erhalten [20].

3. Für die Berechnung der optimalen Politik $\pi^{\text{opt}} = (\pi_0^{\text{opt}}, \dots, \pi_{N-1}^{\text{opt}})$ sind in jedem der N Schritte, für jeden der n Zustände m Entscheidungen zu bewerten. Insgesamt sind es also $N \cdot n \cdot m$ Bewertungen. Bei Bestimmung der optimalen Politik für einen Anfangszustand $x_0 \in X$ durch vollständige Aufzählung wären in jedem der N Schritte m Entscheidungen zu bewerten; insgesamt also für einen Anfangszustand m^N Bewertungen bzw. für alle möglichen Anfangszustände $n \cdot m^N$. Das rekursive Verfahren von Satz 6.2 besitzt offensichtlich einen geringeren Berechnungsaufwand. Für $m = 3$, $n = 3$ und $N = 5$ ist nach nur $N \cdot n \cdot m = 45$ Entscheidungen für jeden Anfangszustand eine optimale Politik bekannt. Bei vollständiger Aufzählung wären für nur einen Anfangszustand $m^N = 243$ Entscheidungen zu betrachten.
4. Der Satz folgt auch unmittelbar aus Satz 6.1, weil für die additive Zielfunktion (6.4) die Funktionen φ_0 und ψ_1 jeweils Summen sind und damit R separierbar und insbesondere φ_0 für jedes $r_0(x_0, u_0)$ im zweiten Argument monoton wachsend ist. Damit ist das *Optimalitätsprinzip* (siehe Seite 242) anwendbar. Das ist die Aussage von Satz 6.2.

6.3.2 Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl

Sei (S, X, U, P, R) ein stochastischer mehrstufiger Entscheidungsprozess mit $N \in \mathbb{N}$ Stufen. Wie im vorigen Abschnitt wird hier wieder von einer *additiven* Zielfunktion ausgegangen. Da bei einem stochastischen Entscheidungsprozess für die Zustandsübergänge nur Übergangswahrscheinlichkeiten $p(x_{t+1} | x_t, u_t)$, $x_t, x_{t+1} \in X$, $u_t \in U$, $t = 0, \dots, N - 1$ bekannt sind, muss der Erwartungswert der Zielfunktion betrachtet werden. Entsprechend (6.6) und (6.7) wird definiert:

$$R_{N-t}^{\max}(x_t) = \max_{u_t, \dots, u_{N-1} \in U} E_{x_t, \dots, x_N | x_{t-1}} R(x_t, u_t, \dots, x_{N-1}, u_{N-1}, x_N) \quad (6.12)$$

für $t = 0, \dots, N-1$ und

$$R_0^{\max}(x_N) = r_N(x_N). \quad (6.13)$$

Dann bezeichnet $R_N^{\max}(x_0)$ den maximalen Erwartungswert der Zielfunktion. Dem Satz 6.2 entspricht in diesem Fall

Satz 6.3: Für (S, X, U, P, R) mit additiver Zielfunktion R wie in (6.4) und $R_{N-t}^{\max}(x_t)$ wie in (6.12) gilt

$$R_{N-t}^{\max}(x_t) = \max_{u_t \in U} [r_t(x_t, u_t) + E_{x_{t+1} | x_t} R_{N-(t+1)}^{\max}(x_{t+1})], \quad t = 0, \dots, N-1 \quad (6.14)$$

und

$$R_0^{\max}(x_N) = g(x_N) \quad (6.15)$$

mit dem bedingten Erwartungswert

$$E_{x_{t+1} | x_t} R_{N-(t+1)}^{\max}(x_{t+1}) = \sum_{x_{t+1} \in X} R_{N-(t+1)}^{\max}(x_{t+1}) p(x_{t+1} | x_t, u_t).$$

□

Beweis: siehe z. B. [16]

Anmerkungen

1. Analog der Berechnung im deterministischen Fall, kann hier das Maximum der Zielfunktion R für einen gegebenen Anfangszustand $x_0 \in X$ sowie eine optimale Politik π berechnet werden (vgl. Anmerkungen zu Satz 6.2 auf Seite 250).
2. Der Satz folgt auch unmittelbar aus Satz 6.1, weil für die additive Zielfunktion (6.4) die Funktionen φ_0 und ψ_1 jeweils Summen sind und damit R separierbar und insbesondere φ_0 für jedes $r_0(x_0, u_0)$ im zweiten Argument monoton wachsend ist. Damit ist das *Optimalitätsprinzip* (siehe Seite 242) anwendbar. Das ist die Aussage von Satz 6.3.

6.3.3 Stochastischer mehrstufiger Entscheidungsprozess mit fester Stufenzahl in unscharfer Umgebung

Die Grundlagen für die Erweiterung der dynamischen Optimierung für unscharfe Umgebungen wurden in Abschnitt 6.2 dargestellt. Hier wird zunächst Def. 6.2 für eine Variante unscharfer Umgebungen erweitert und das zugehörige Optimierungsproblem formuliert.

Definition 6.9: *Mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit¹¹⁾.*

Sei (S, X, U, P) ein mehrstufiger stochastischer Entscheidungsprozess gemäß Def. 6.2, jedoch ohne Zielfunktion R . Die Prozess-Endzeit $N \in \mathbb{N}$, das heißt die Anzahl der Stufen sei fest vorgegeben. Für jede Entscheidung $u_t \in U$ gebe es eine unscharfe Nebenbedingung $C_t \in \mathcal{IF}(U)$ und für den stochastischen Endzustand $x_N \in X$ ein unscharfes Ziel $G_N \in \mathcal{IF}(X)$ ¹²⁾. Dann bezeichnet (S, X, U, P, C, G, N) einen *mehrstufigen stochastischen Entscheidungsprozess in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit N* . \square

Mit dem Aggregationsoperator \wedge (Minimum-Operator) und einem Anfangszustand $x_0 \in X$, der durch eine Folge von Entscheidungen u_0, \dots, u_{N-1} (nicht deterministisch) in einen Endzustand $x_N \in X$ überführt wird, bezeichnet

$$\mu_D(u_0, \dots, u_{N-1} | x_0) := \mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge \mu_{G_N}(x_N) \quad (6.16)$$

die *unscharfe Entscheidung*. Wegen der stochastischen Zustandsübergänge ist (6.16) (für gegebenen Anfangszustand und Entscheidungen) eine Zufallsvariable. Es liegt deshalb nahe, das Optimierungsproblem wie Kacprzyk und Staniewski [84, 86] zu formulieren:

Finde eine (optimale) Folge von Entscheidungen $u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}$, die den Erwartungswert der unscharfen Entscheidung für den Anfangszustand $x_0 \in X$ maximiert, das heißt

$$\mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} | x_0) := \max_{u_0, \dots, u_{N-1}} E[\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge \mu_{G_N}(x_N)]. \quad (6.17)$$

Dieser Ansatz stammt aus dem Jahre 1980. Die ältere und erste Formulierung des Problems von Bellman und Zadeh [11] lautet:

Finde eine (optimale) Folge von Entscheidungen $u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}$, die den Erwartungswert des unscharfen Ziels unter den gegebenen unscharfen Nebenbedingungen für den Anfangszustand $x_0 \in X$ maximiert, das heißt

$$\mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} | x_0) := \max_{u_0, \dots, u_{N-1}} (\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge E[\mu_{G_N}(x_N)]). \quad (6.18)$$

11. N -stufige Entscheidungsprozesse haben die Prozess-Endzeit N . Es ist also gleichgültig, ob von Entscheidungsprozessen mit fester Stufen-Anzahl oder von Entscheidungsprozessen mit fest vorgegebener Prozess-Endzeit gesprochen wird.

12. Der Index N verdeutlicht dass das unscharfe Ziel den Endzustand bewertet. Formal wäre der Index nicht nötig.

Da x_N von x_0 und u_0, \dots, u_{N-1} abhängt, ist der Erwartungswert bei ausführlicher Schreibweise für (6.17):

$$\begin{aligned} & E[\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge \mu_{G_N}(x_N)] \\ &= E_{x_N | (x_0, u_0, \dots, u_{N-1})}[\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge \mu_{G_N}(x_N)] \end{aligned}$$

und für (6.18):

$$E[\mu_{G_N}(x_N)] = E_{x_N | (x_0, u_0, \dots, u_{N-1})}[\mu_{G_N}(x_N)]$$

(siehe auch die Anmerkungen auf Seite 255). Aus folgenden Gründen wird im Weiteren der zweite Ansatz zugrunde gelegt:

- Der Ansatz von Bellman und Zadeh ist in der Literatur verbreiteter und in Anwendungen gebräuchlicher.
- Auch er behandelt – wie weiter unten deutlich wird – die Stochastik in mehrstufigen stochastischen Entscheidungsprozessen, (S, X, U, P, C, G, N) in „plausibler“ Weise.
- Er fügt sich, im Gegensatz zum ersten Ansatz, besser in die aus Abschnitt 6.3.1 bekannte Schematik ein (was nicht verwundert, weil diese Schematik auf Bellman zurückgeht)¹³⁾.

Das Optimierungsproblem bestehe nun also darin, für einen Anfangszustand x_0 eine optimale Folge von Entscheidungen $u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}$ zu finden, bzw. eine optimale Politik

$$\pi^{\text{opt}}(x_0) = (\pi_0^{\text{opt}}(x_0), \dots, \pi_{N-1}^{\text{opt}}(x_{N-1}))$$

mit

$$\pi_t^{\text{opt}} : X \rightarrow U \text{ und } u_t^{\text{opt}} = \pi_t^{\text{opt}}(x_t), \quad (6.19)$$

so dass:

$$\begin{aligned} & \mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} | x_0) \\ &= \max_{u_0, \dots, u_{N-1}} (\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge E_{x_N | (x_0, u_0, \dots, u_{N-1})} \mu_{G_N}(x_N)) \end{aligned} \quad (6.20)$$

Die Bedeutung des Terms in (6.20) erschließt sich bei der folgenden Berechnung des Erwartungswertes

$$E_{x_N | (x_0, u_0, \dots, u_{N-1})} \mu_{G_N}(x_N).$$

Mit dem bedingten Erwartungswert

13. Iwamoto drückt es so aus: „Then for stochastic process they (Bellman und Zadeh, d. V.) have just replaced formally the recursive equation (des deterministischen Falls, d. V.) with a stochastic version through a straightforward analogy.“ [80]

$$E_{x_N | (x_{N-1}, u_{N-1})} \mu_{G_N}(x_N) = \sum_{x_N \in X} p(x_N | x_{N-1}, u_{N-1}) \mu_{G_N}(x_N) \quad (6.21)$$

ist

$$\begin{aligned} \mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} | x_0) &= \max_{u_0, \dots, u_{N-1}} \left[\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \right. \\ &\quad \left. \wedge E_{x_{N-1} | (x_0, u_0, \dots, u_{N-2})} \left[\sum_{x_N \in X} p(x_N | x_{N-1}, u_{N-1}) \mu_{G_N}(x_N) \right] \right] \end{aligned}$$

Weil die beiden Terme ganz rechts nur von u_{N-1} abhängen und weil die Minimumbildung monoton wachsend ist, kann geschrieben werden:

$$\begin{aligned} \mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} | x_0) \\ = \max_{u_0, \dots, u_{N-2}} (\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-2}}(u_{N-2}) \wedge E_{x_{N-1} | (x_0, u_0, \dots, u_{N-2})} \mu_{G_{N-1}}(x_{N-1})) \end{aligned}$$

wobei $\mu_{G_{N-1}}(x_{N-1}) = \max_{u_{N-1}} (\mu_{C_{N-1}}(u_{N-1}) \wedge E_{x_N | (x_{N-1}, u_{N-1})} \mu_{G_N}(x_N))$. Die Wiederholung dieser Rückwärts-Iteration führt zu folgenden Rekursionsgleichungen:

$$\mu_{G_{N-i}}(x_{N-i}) := \max_{u_{N-i}} (\mu_{C_{N-i}}(u_{N-i}) \wedge E_{x_{N-i+1} | (x_{N-i}, u_{N-i})} \mu_{G_{N-i+1}}(x_{N-i+1})) , \quad (6.22)$$

$$\begin{aligned} E_{x_{N-i+1} | (x_{N-i}, u_{N-i})} \mu_{G_{N-i+1}}(x_{N-i+1}) \\ = \sum_{x_{N-i+1} \in X} p(x_{N-i+1} | x_{N-i}, u_{N-i}) \mu_{G_{N-i+1}}(x_{N-i+1}) \end{aligned} \quad (6.23)$$

für $i = 1, \dots, N$. Damit ist (6.20) gleichbedeutend mit

$$\max_{u_0, \dots, u_{N-1}} E_{x_1 | (x_0, u_0)} [\mu_{C_0}(u_0) \wedge E_{x_2 | (x_1, u_1)} (\mu_{C_1}(u_1) \wedge \dots \wedge E_{x_N | (x_{N-1}, u_{N-1})} \mu_{G_N}(x_N)) \dots] .$$

Durch rekursives Lösen der Gleichungen (6.22) und (6.23) folgen nacheinander u_{N-i}^{opt} bzw. die optimalen Politiken π_{N-i}^{opt} so dass $u_{N-i}^{\text{opt}} = \pi_{N-i}^{\text{opt}}(x_{N-i})$. Wie bei den scharfen Entscheidungsprozessen in den Abschnitten 6.3.1 und 6.3.2 kommt hier das Rekursionsprinzip zu Anwendung.

Anmerkungen

1. Weil im Allgemeinen die Gleichung

$$E[\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge \mu_{G_N}(x_N)] = \mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge E[\mu_{G_N}(x_N)]$$

nicht gilt, ergeben die Ansätze von Bellman/Zadeh und Kacprzyk/Staniewski in der Regel unterschiedliche Ergebnisse.

2. Der Erwartungswert (6.21) wird bei Zadeh als (scharfe) Wahrscheinlichkeit des (unscharfen) Ereignisses G_N betrachtet:

$$P\{G_N\} = E\mu_{G(N)} .$$

Auch zur Berechnung von Wahrscheinlichkeiten unscharfer Ereignisse gibt es Alternativen (siehe [86]). Sie sind in der Anwendung wenig gebräuchlich und werden deshalb hier nicht weiter betrachtet.

- Der Ansatz von Kacprzyk/Staniewski impliziert eine Politik, die nicht nur vom aktuellen Zustand abhängt, sondern eine Abbildung

$$\pi : [0, 1] \times X \rightarrow U$$

ist, die den aktuellen Zustand und die in einem Wert im Intervall $[0, 1]$ „zusammengefasste Vorgeschichte“ auf eine Entscheidung abbildet¹⁴⁾ [84, 86].

- Weitere Ansätze (die dem von Kacprzyk/Staniewski ähnlich sind) finden sich bei Iwamoto [80, 81].

Der folgende Satz fasst oben Gesagtes zusammen.

Satz 6.4: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit N gemäß Def. 6.9, (S, X, U, P, C, G, N) . Für einen Anfangszustand $x_0 \in X$ sei

$$\pi^{\text{opt}}(x_0) = (\pi_0^{\text{opt}}(x_0), \dots, \pi_{N-1}^{\text{opt}}(x_{N-1}))$$

mit $\pi_t^{\text{opt}} : X \rightarrow U$ eine optimale Politik gemäß der Formulierung von Bellman und Zadeh, das heißt für $u_t^{\text{opt}} = \pi_t^{\text{opt}}(x_t)$ gilt

$$\begin{aligned} & \mu_D(u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}} \mid x_0) \\ &= \max_{u_0, \dots, u_{N-1}} (\mu_{C_0}(u_0) \wedge \dots \wedge \mu_{C_{N-1}}(u_{N-1}) \wedge E_{x_N \mid (x_0, u_0, \dots, u_{N-1})} \mu_{G_N}(x_N)) \end{aligned}$$

Dann lässt sich diese optimale Politik bzw. die Folge optimaler Entscheidungen $u_0^{\text{opt}}, \dots, u_{N-1}^{\text{opt}}$ durch rekursives Lösen der Gleichungen:

$$\mu_{G_{N-i}}(x_{N-i}) = \max_{u_{N-i}} (\mu_{C_{N-i}}(u_{N-i}) \wedge E_{x_{N-i+1}} \mu_{G_{N-i+1}}(x_{N-i+1})) ,$$

mit dem bedingten Erwartungswert

14. Hinweise auf weitere Ansätze zur Lösung eines mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit sind ebenfalls in [86] zu finden.

$$E_{x_{N-i+1} | x_{N-i}} \mu_{G_{N-i+1}}(x_{N-i+1}) := \sum_{x_{N-i+1} \in X} p(x_{N-i+1} | x_{N-i}, u_{N-i}) \mu_{G_{N-i+1}}(x_{N-i+1})$$

für $i = 1, \dots, N$ berechnen.

□

Anmerkung

Die Aussage folgt auch aus Satz 6.1, weil die Zielfunktion trennbar und die Minimumbildung in jedem Argument monoton wachsend ist. Damit ist das *Optimalitätsprinzip* (siehe Seite 242) anwendbar.

6.4 Dynamische Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit

Im vorigen Abschnitt wird die Methodik der dynamischen Optimierung mehrstufiger stochastischer Entscheidungsprozesse mit fester Stufenzahl eingeführt. Die Modelle mit fester Stufenzahl sind aber nicht geeignet die im Kapitel 5 beschriebene modellbasierte Kundenakquisition auf der Grundlage dynamischer Akquisepläne abzubilden. Obgleich die Anzahl der Akquiseaktionen praktisch beschränkt ist, entspricht es mehr der Realität, das Ende eines Akquiseprozesses nicht über das Erreichen einer vorgegebenen Stufenzahl, das heißt eines vorgegebenen Zeitschritts zu beschreiben, sondern über das Erreichen eines Endzustands. In der Herleitung der Def. 5.9 – dynamische Akquisepläne – ergibt sich diese Modellierung wie von selbst aus praktischen Erwägungen. Somit geben der Endzustand bzw. die Menge der Endzustände implizit die Prozess-Endzeit vor.

Die Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit wurde in der Literatur bisher nicht ausführlich behandelt¹⁵). Nachfolgend wird der Ansatz aus [161, 163, 164] vollständig ausgearbeitet und genau beschrieben. Im ersten Abschnitt wird der Entscheidungsprozess definiert und das Optimierungsproblem in die Gestalt einer Funktionalgleichung überführt. Im Abschnitt 6.4.2 wird ein notwendiges und hinreichendes Kriterium für die Existenz von zulässigen Politiken hergeleitet und gezeigt, wie solche Politiken berechnet werden können. Davon ausgehend zeigt Abschnitt 6.4.3 die Existenz einer optimalen Politik und wie sie iterativ berechnet werden kann.

15. Solche Systeme werden auch in den beiden Monographien [84, 86] nicht behandelt. Dort finden sich nur folgende Anmerkungen: „An extension into the stochastic system under control would imply some difficulties even in the phase of problem formulation, not to speak about the solution“ [a. a. O. S. 63] bzw. „Concerning the case of a stochastic system under control, one encounters difficulties even while trying to formally state the control problem“ [a. a. O. S. 188].

6.4.1 Grundlagen

Definition 6.10: *Mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, Menge der Endzustände, Menge der nicht-beendeten Zustände.*

Ein *mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit* ist ein dynamisches System (S, X) mit der Zustandsmenge

$$X = \{\sigma_1, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n\}, \quad (6.24)$$

mit diskreten Zuständen

$$x_t \in X, t = 0, 1, 2, \dots$$

und mit den folgenden Eigenschaften:

Die Teilmengen

$$T = \{\sigma_{k+1}, \dots, \sigma_n\} \quad (6.25)$$

und

$$\bar{T} = X \setminus T = \{\sigma_1, \dots, \sigma_k\} \quad (6.26)$$

bezeichnen die *Menge der Endzustände* („termination set“) T bzw. die *Menge der nicht-beendeten Zustände* („non-terminated states“) \bar{T} . Es wird angenommen, dass der Anfangszustand kein Endzustand ist, das heißt

$$x_0 \in \bar{T}.$$

Der Entscheidungsprozess ist beendet, wenn erstmals ein Endzustand erreicht wird.

Zu jedem Zustand $x_t \in X$ gibt es eine unscharfe Nebenbedingung $C(x_t) \in \mathcal{IF}(U)$ über der Entscheidungsmenge U . Für ihre Zugehörigkeitsfunktionen gilt:

$$\mu_{C(x_t)}(u_t) \begin{cases} \geq 0 & \text{for } x_t \in \bar{T} \\ = 0 & \text{for } x_t \in T \end{cases} \quad (6.27)$$

für beliebige Entscheidungen $u_t \in U$ ¹⁶⁾.

16. Im Unterschied zu den unscharfen Nebenbedingungen der Entscheidungsprozesse im Abschnitt 6.3, die von der Zeit t abhängig sind, ist es bei implizit vorgegebener Prozess-Endzeit zweckmäßig, die Nebenbedingungen in Abhängigkeit vom Zustand zu formulieren. Das entspricht auch den praktischen Notwendigkeiten bei der modellbasierten Neukundengewinnung (siehe Kapitel 5, Def. 5.7).

Für den Entscheidungsprozess gibt es ein unscharfes Ziel $G \in \mathcal{IF}(X)$ über der Zustandsmenge X ¹⁷⁾. Für seine Zugehörigkeitsfunktion gilt:

$$\mu_G(x_t) \begin{cases} \geq 0 & \text{for } x_t \in T \\ = 0 & \text{for } x_t \in \bar{T} \end{cases} \quad 18). \quad (6.28)$$

Zu jeder Entscheidung $u_t \in U$ gibt es eine stochastische Matrix $P(u_t)$ mit den Übergangswahrscheinlichkeiten zwischen Folgezuständen für diese Entscheidung. Für die bedingten Wahrscheinlichkeiten eines Zustandsübergangs bei gegebener Entscheidung $u_t \in U$ gilt

$$p(x_{t+1} | x_t, u_t) \begin{cases} \geq 0 & \text{für } x_t \in \bar{T} \\ = 0 & \text{für } x_t \in T, x_{t+1} \neq x_t \\ = 1 & \text{für } x_t = x_{t+1} \in T \end{cases} \quad (6.29)$$

für jedes $u_t \in U$.

Ein so definierter mehrstufiger Entscheidungsprozess wird mit dem Tupel (S, X, T, U, P, C, G) bezeichnet. \square

Bei einem stochastischen System mit implizit definierter Endzeit ist die Anzahl der Zustände vom Anfangszustand bis zu einem Endzustand nicht im Voraus bekannt. Vielmehr ist die Anzahl der Schritte N vom Anfangszustand $x_0 \in \bar{T}$ zu einem Endzustand $x_N \in T$ eine Zufallsgröße, die von x_0 , der Folge der Eingabewerte u_0, u_1, \dots, u_{N-1} und den entsprechenden bedingten Übergangswahrscheinlichkeiten $p(x_{t+1} | x_t, u_t)$ abhängt. Für die unscharfe Entscheidung wird geschrieben:

17. Das bedeutet, alle Zustände der Zustandsmenge werden bewertet, auch die nicht-beendeten Zustände.

18. Im Unterschied zu den unscharfen Zielen der Entscheidungsprozesse im Abschnitt 6.3, die von der Zeit t abhängig sind (für jedes $t = 0, 1, \dots, N$ oder nur für $t = N$) wird hier nur ein Ziel für den Endzustand des Systems formuliert. Daher hätte es genügt, G nur über T zu definieren. Durch $G \in \mathcal{IF}(X)$ wird aber die Darstellung im Folgenden erleichtert. Mit der Wahl der Zugehörigkeitswerte für die Endzustände ist es möglich, zwischen „erwünschten“ (hoher Zugehörigkeitswert), „weniger erwünschten“ (mittlerer Zugehörigkeitswert) und „eher unerwünschten“ (geringer Zugehörigkeitswert) Endzuständen zu unterscheiden. Das entspricht auch der Anwendung bei der modellbasierten Neukundengewinnung.

$$\begin{aligned} \mu_D(u_0, \dots, u_{N-1} \mid x_0) = \\ \mu_{C(x_0)}(u_0) \wedge E_{x_1 \mid (x_0, u_0)} [\mu_{C(x_1)}(u_1) \wedge E_{x_2 \mid (x_1, u_1)} (\mu_{C(x_2)}(u_2) \wedge \dots \\ \dots \wedge E_{x_{N-1} \mid (x_{N-2}, u_{N-2})} [\mu_{C(x_{N-1})}(u_{N-1}) \wedge E_{x_N \mid (x_{N-1}, u_{N-1})} \mu_G(x_N)] \dots)] \end{aligned} \quad (6.30)$$

wobei die Anzahl der Schritte bis zum Endzustand $x_N \in T$, N , und die Zustände $x_1, \dots, x_{N-1} \in \bar{T}$ Zufallsgrößen sind. Hier und im Folgenden ist der Minimum-Operator der alleinige Aggregationsoperator¹⁹⁾. Diese Wahl folgt dem ursprünglichen Ansatz von Bellman und Zadeh [11]. Eine allgemeine Betrachtung des Aggregationsoperators gibt Anmerkung 1 zu Bsp. 6.1. Und im Abschnitt 6.4.4 wird gezeigt für welche anderen t-Normen die Aussagen dieses Abschnitts verallgemeinert werden können.

Für die Überführung der unscharfen Entscheidung (6.30) in eine rekursive Form wird im Folgenden angenommen, dass die aufeinanderfolgenden Eingabewerte $u_0, u_1, \dots, u_t, \dots$ das Ergebnis einer stationären Politik $\pi : \bar{T} \rightarrow U$ sind, das heißt

$$u_t = \pi(x_t), t = 0, 1, 2, \dots$$

Für die linke Seite von (6.30) kann dann kürzer

$$\mu_D(\pi \mid x_0)$$

geschrieben werden sowie

$$E_{x_{t+1} \mid (x_t, u_t)} = E_{x_{t+1} \mid (x_t, \pi(x_t))} =: E_{x_{t+1} \mid x_t},$$

und für die rechte Seite folgt dann

$$\begin{aligned} \mu_{C(x_0)}(\pi(x_0)) \\ \wedge E_{x_1 \mid x_0} [\mu_{C(x_1)}(\pi(x_1)) \wedge \dots \wedge E_{x_{N-1} \mid x_{N-2}} (\mu_{C(x_{N-1})}(\pi(x_{N-1})) \wedge E_{x_N \mid x_{N-1}} \mu_G(x_N)) \dots] \end{aligned}$$

Wird für beliebige Politiken π

$$\begin{aligned} \mu_D(\pi \mid x_t) &:= \mu_{C(x_t)}(\pi(x_t)) \\ &\wedge E_{x_{t+1} \mid x_t} [\mu_{C(x_{t+1})}(\pi(x_{t+1})) \wedge \dots \wedge E_{x_{N-1} \mid x_{N-2}} (\mu_{C(x_{N-1})}(\pi(x_{N-1})) \wedge E_{x_N \mid x_{N-1}} \mu_G(x_N)) \dots] \end{aligned}$$

für $x_t \in \bar{T}$ und

19. In [12] zeigen Giertz und Bellman, dass unter bestimmten „vernünftigen“ Annahmen, nur die Minimumsbildung als Aggregationsoperator möglich ist.

$$\mu_D(\pi \mid x_t) := \mu_G(x_t) \quad (6.31)$$

für $x_t \in T$ sowie

$$\mu_C(\pi(x_t) \mid x_t) := \mu_{C(x_t)}(\pi(x_t)), \quad (6.32)$$

gesetzt, dann ergibt sich die Gleichung

$$\mu_D(\pi \mid x_t) = \mu_C(\pi(x_t) \mid x_t) \wedge E_{x_{t+1} \mid x_t} \mu_D(\pi \mid x_{t+1}) . \quad (6.33)$$

Für $t = 0$ wird aus (6.33)

$$\mu_D(\pi \mid x_0) = \mu_C(\pi(x_0) \mid x_0) \wedge E_{x_1 \mid x_0} \mu_D(\pi \mid x_1) ,$$

was der ursprünglichen Formulierung (6.30) entspricht.

Die Mengen mit den Zugehörigkeitsfunktionen (6.31) und (6.32) seien mit $D_\pi(x_t)$ und $C_\pi(x_t)$ bezeichnet. Die Rekursionsgleichung (6.33) lässt sich so interpretieren: Für eine bestimmte Politik π bei einem gegebenen Zustand $x_t \in \bar{T}$ ist die unscharfe Entscheidung $D_\pi(x_t)$ bezüglich der Politik π das Zusammenkommen von der Nebenbedingung im Zustand x_t und der zu erwartenden unscharfen Entscheidung im Folgezustand x_{t+1} , $D_\pi(x_{t+1})$, bezüglich der Politik π .

Für den Erwartungswert in (6.33) gilt

$$\begin{aligned} E_{x_{t+1} \mid x_t} \mu_D(\pi \mid x_{t+1}) &= \sum_{x_{t+1} \in \bar{T}} p(x_{t+1} \mid x_t, \pi(x_t)) \cdot \mu_D(\pi \mid x_{t+1}) \\ &+ \sum_{x_{t+1} \in T} p(x_{t+1} \mid x_t, \pi(x_t)) \cdot \mu_G(x_{t+1}) \end{aligned}$$

und damit erhält (6.33) die Form

$$\begin{aligned} \mu_D(\pi \mid x_t) &= \mu_C(\pi(x_t) \mid x_t) \wedge \left[\sum_{x_{t+1} \in \bar{T}} p(x_{t+1} \mid x_t, \pi(x_t)) \cdot \mu_D(\pi \mid x_{t+1}) \right. \\ &\quad \left. + \sum_{x_{t+1} \in T} p(x_{t+1} \mid x_t, \pi(x_t)) \cdot \mu_G(x_{t+1}) \right] . \end{aligned} \quad (6.34)$$

Das ist ein System von n Gleichungen (eine Gleichung für jeden Zustand, den x_t annehmen kann), wobei n die Anzahl der Elemente von X ist. Wegen (6.31) reicht es aber, nur die k

Zustände in \bar{T} zu betrachten. Zur Vereinfachung von Gleichungssystem (6.34) werden folgende Vektoren eingeführt.

Definition 6.11: *Politikvektor, Vektor der unscharfen Nebenbedingungen, unscharfer Entscheidungsvektor, unscharfer Zielvektor.*

Der *Politikvektor* $(\pi(\sigma_1), \dots, \pi(\sigma_k))^T$ repräsentiert die stationäre Politik $\pi : X \rightarrow U$. Er enthält die Bilder aller nicht-beendeten Zustände $\sigma_1, \dots, \sigma_k \in \bar{T}$ und wird mit der Politik identifiziert, das heißt $\pi \triangleq (\pi(\sigma_1), \dots, \pi(\sigma_k))^T$. $\mu_C(\pi) = (\mu_C(\pi(\sigma_1) | \sigma_1), \dots, \mu_C(\pi(\sigma_k) | \sigma_k))^T$ ist der *Vektor der unscharfen Nebenbedingungen*, $\mu_D(\pi) = (\mu_D(\pi | \sigma_1), \dots, \mu_D(\pi | \sigma_k))^T$ ist der *unscharfe Entscheidungsvektor* und $\mu_G = (\mu_G(\sigma_{k+1}), \dots, \mu_G(\sigma_n))^T$ ist der *unscharfe Zielvektor*. Die Komponenten von $\mu_C(\pi)$ und $\mu_D(\pi)$ sind die Werte der Zugehörigkeitsfunktionen von C_π und D_π für $\sigma_1, \dots, \sigma_k$. Die Komponenten von μ_G sind die Werte der Zugehörigkeitsfunktion des unscharfen Zieles G für die Endzustände in T . \square

Werden für eine gegebene Politik π die Übergangswahrscheinlichkeiten aus (6.29) für $x_t, x_{t+1} \in X$ und für $u_t = \pi(x_t)$ in einer Matrix $P(\pi) \in [0, 1]^{n \times n}$ zusammengefasst, so besitzt diese Übergangsmatrix die Gestalt

$$P(\pi) := (p_{ij}(\pi))_{i,j} = \begin{pmatrix} P_{\bar{T}}(\pi) & P_T(\pi) \\ 0 & E \end{pmatrix} \in [0, 1]^{n \times n}, \quad (6.35)$$

wobei

$$p_{ij}(\pi) := p(\sigma_j | \sigma_i, \pi(\sigma_i)) \quad (6.36)$$

für $i, j = 1, \dots, n$ die Übergangswahrscheinlichkeit vom Zustand σ_i zum Zustand σ_j unter der Politik π ist. Die $k \times k$ -Untermatrix

$$P_{\bar{T}}(\pi) := (p_{ij}(\pi))_{i,j=1,\dots,k} \quad (6.37)$$

enthält die Übergangswahrscheinlichkeiten zwischen nicht-beendeten Zuständen der Menge \bar{T} und die $k \times (n - k)$ -Übergangsmatrix

$$P_T(\pi) := (p_{ij}(\pi))_{\substack{i=1,\dots,k \\ j=k+1,\dots,n}} \quad (6.38)$$

enthält die Übergangswahrscheinlichkeiten von den nicht-beendeten Zuständen in \bar{T} zu beendeten Zuständen in T , jeweils bei gegebener Politik π . E ist eine $(n - k) \times (n - k)$ -Einheitsmatrix, das heißt die Übergangswahrscheinlichkeiten zwischen verschiedenen beendeten Zuständen sind gleich null, so dass der Prozess nach Eintritt in einen beendeten Zustand dort verharret. Deshalb ist auch kein Übergang von einem beendeten zu einem nicht-beendeten Zustand möglich, was durch die Nullmatrix gekennzeichnet wird.

Anmerkungen²⁰⁾

1. Für eine gegebene Politik π hängt die Übergangswahrscheinlichkeit $p(x_{t+1} | x_t, \pi(x_t))$ zwischen zwei beliebigen Zuständen $x_t, x_{t+1} \in X$ alleine von den Zuständen ab, insbesondere ist sie von der Zeit t unabhängig. Deshalb bildet der hier betrachtete mehrstufige Entscheidungsprozess für jede Politik π eine *stationäre Markov-Kette*.
2. Für eine gegebene Politik $\pi \in \Pi$ gilt:
 - Ist ein Endzustand aus $T = \{\sigma_{k+1}, \dots, \sigma_n\}$ erreicht, kann er wegen (6.29) nicht mehr verlassen werden, das heißt jeder Endzustand ist *absorbierend* („absorbing“) und die Menge T *abgeschlossen* („closed“).
 - Wenn \bar{T} keine abgeschlossene Teilmenge enthält (und auch selbst nicht abgeschlossen ist), dann liegt die Übergangsmatrix (6.35) in *kanonischer Form* vor. Insbesondere gilt dann für alle nicht beendeten Zustände aus \bar{T} , dass ein Endzustand aus T erreichbar ist, das heißt es gibt ein $\kappa \in \mathbb{N}$, so dass $p_{ij}^{(\kappa)}(\pi) > 0$, wobei $p_{ij}^{(\kappa)}(\pi)$ die κ -Schritt Übergangswahrscheinlichkeit bezeichnet²¹⁾. Dann gibt es in X nur die abgeschlossenen Zustandsmengen $\{\sigma_i\}$, $i = k + 1, \dots, n$.
 - Wenn die Übergangsmatrix kanonisch ist und es genau einen Endzustand gibt, dann ist die Markov-Kette *einkettig* („unichain“), bei mehr als einem Endzustand ist sie *mehrkettig* („multi chain“).
 - Auf die Abhängigkeit der Eigenschaften der Menge \bar{T} und ihrer Elemente von der Politik π wird weiter unten eingegangen.
3. In Def. 6.10 werden die stochastischen Matrizen $P(u_t)$, $u_t \in U$ als Bestandteile eines mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit eingeführt. Sie besitzen für gegebenen Zustand $x_t \in X$ und gegebene Entscheidung $u_t \in U$ die Eigenschaft

$$\sum_{x_{t+1} \in X} p(x_{t+1} | x_t, u_t) = 1. \quad (6.39)$$

20. vgl. [134]

21. Die κ -Schritt Übergangswahrscheinlichkeit $p_{ij}^{(\kappa)}(\pi)$ ist das Element (i, j) der Matrix $P^{(\kappa)}(\pi)$.

Die Matrizen $P(\pi)$, $\pi \in \Pi$ sind ebenfalls stochastische Matrizen, denn mit $u_t = \pi(x_t)$ und (6.39) gilt für gegebenen Zustand $x_t \in X$:

$$\sum_{x_{t+1} \in X} p(x_{t+1} | x_t, \pi(x_t)) = \sum_{x_{t+1} \in X} p(x_{t+1} | x_t, u_t) = 1.$$

Mit den Vektoren aus Def. 6.11 und den Matrizen aus (6.37) und (6.38) erhält das Gleichungssystem (6.34) die Gestalt:

$$\mu_D(\pi) = \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \mu_D(\pi) + P_T(\pi) \cdot \mu_G], \quad (6.40)$$

wobei der Minimum-Operator \wedge komponentenweise auf die Vektoren angewandt wird. Aus der Gleichung ist ersichtlich, dass der Entscheidungsvektor zur Politik π , $\mu_D(\pi)$, ein *Fixpunkt* der Transformation

$$\begin{aligned} T_\pi : [0, 1]^k &\rightarrow [0, 1]^k \\ T_\pi(\omega) &:= \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \omega + P_T(\pi) \cdot \mu_G] \end{aligned} \quad (6.41)$$

ist.

Mit Hilfe der in diesem Abschnitt gelegten Grundlagen, und insbesondere der Funktionalgleichung (6.40) sollen die folgenden **Fragen** beantwortet werden:

1. Gibt es eine Politik π , die das System für jeden Anfangszustand mit Wahrscheinlichkeit eins in einen Endzustand überführt? Wodurch zeichnet sie sich aus?
2. Wie kann zu einer gegebenen Politik π der Entscheidungsvektor $\mu_D(\pi)$ berechnet werden, das heißt wie kann eine Politik bewertet werden?
3. Wie wird eine optimale Politik π^{opt} definiert und unter welchen Voraussetzungen existiert sie?
4. Wie kann gegebenenfalls π^{opt} berechnet werden?

Die Fragen 1 – 4 werden in den beiden folgenden Unterabschnitten beantwortet. Für die Klärung der Fragen erweist sich die Fixpunkteigenschaft der Entscheidungsvektoren als besonders hilfreich. Auf sie wird an entsprechender Stelle näher eingegangen.

6.4.2 Zulässige Politiken

Bei der Betrachtung von mehrstufigen stochastischen Entscheidungsprozessen mit implizit vorgegebener Prozess-Endzeit ist die „Erreichbarkeit“ eines Endzustands wesentlich, weil darüber das Prozessende definiert wird. Die folgende Definition klärt in Anlehnung an [134] was damit gemeint ist und stellt so den ersten Schritt zur Klärung von **Frage 1** dar.

Definition 6.12: Erreichbarkeit unter der Politik π , Zulässige Politik, Menge der zulässigen Politiken.

Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung (S, X, T, U, P, C, G) nach Def. 6.10. Ein Zustand $\sigma_j \in X$ heißt *erreichbar* vom Zustand $\sigma_i \in X$ unter der Politik π , $i, j \in \{1, 2, \dots, n\}$, wenn es ein $K = K(\sigma_i, \sigma_j) \in \mathbb{N}_0$ gibt, so dass für die K -Schritt-Übergangswahrscheinlichkeit

$$p_{ij}^{(K)}(\pi) := p^{(K)}(\sigma_j | \sigma_i, \pi) > 0 \quad (6.42)$$

gilt²²⁾. Eine Politik $\pi \in \Pi$ ist *zulässig*, wenn aus jedem (nicht beendeten) Anfangszustand des Entscheidungsprozesses ein Endzustand erreichbar ist. Die Menge der zulässigen Politiken wird mit Π_p bezeichnet. \square

Anmerkungen

1. Nach den Chapman-Kolmogorow Gleichungen [155] ist $p_{ij}^{(K)}(\pi)$ die K -Schritt-Übergangswahrscheinlichkeit $p(x_{t+K} = \sigma_j | x_t = \sigma_i, \pi)$. Daher bedeutet die Erreichbarkeit des Zustandes σ_j vom Zustand σ_i unter der Politik π , dass mit der Politik π der Zustand σ_j vom Zustand σ_i in K Zeitschritten mit positiver Wahrscheinlichkeit erreicht werden kann.
2. Wenn σ_i ein nicht beendeter Zustand und σ_j ein Endzustand ist, das heißt $\sigma_i \in \bar{T}$ und $\sigma_j \in T$, und wenn σ_j unter der Politik π vom Zustand σ_i erreichbar ist, dann gibt es ein $K \leq k := |\bar{T}|$ mit einer positiven K -Schritt Übergangswahrscheinlichkeit (6.42). Das folgt unmittelbar aus der Endlichkeit der Menge \bar{T} .
3. Aus der Zulässigkeit einer Politik folgt nicht, dass jeder Endzustand von einem nicht beendeten Zustand erreichbar ist.

Beispiel 6.2: Der Satz 6.6 unten zeigt, wie die Zulässigkeit einer Politik anhand der Eigenschaften der Matrix $P(\pi)$ aus (6.35) überprüft werden kann und im Korollar 6.6.1 wird die Erreichbarkeit der Menge der beendeten Zustände genauer bestimmt. Für ihren Beweis ist folgender Hilfssatz nötig.

Hilfssatz 6.5: Gegeben sei die stochastische Matrix $P(\pi)$ aus (6.35). Dann ist P^K für $K = 0, 1, 2, \dots$ auch eine stochastische Matrix und es gilt

22. In der englischsprachigen Fachliteratur wird diese Eigenschaft als „reachable“ oder „accessible“ bezeichnet. Vgl. zum Beispiel [134].

$$P^K(\pi) = \begin{pmatrix} P_{\bar{T}}^K(\pi) & B(\pi) \\ 0 & E \end{pmatrix} := (p_{ij}^{(K)}(\pi))_{i,j} \quad (6.43)$$

wobei $B(\pi)$ eine Funktion von $P_{\bar{T}}(\pi)$ und $P_T(\pi)$ ist. \square

Beweis: Der Beweis kann durch vollständige Induktion geführt werden. Per Definition sei die nullte Potenz einer quadratischen Matrix gleich der Einheitsmatrix entsprechender Größe.

1. Für $K = 0$ ist per Definition $P^K(\pi) = E_{n \times n}$, das heißt $P_{\bar{T}}^K(\pi) = E_{k \times k}$ und $B(\pi) = 0_{k \times (n-k)}$ ²³⁾. Damit gilt (6.43) und $P^K(\pi)$ ist offensichtlich stochastisch.
2. Es wird angenommen, dass für ein beliebiges aber festes $K \in \mathbb{N}$, $K > 1$ (6.43) gilt und P^K eine stochastische Matrix ist.
3. Unter Nutzung der Induktionsannahme 2. ist

$$P^{K+1} = P \cdot P^K = \begin{pmatrix} P_{\bar{T}}(\pi) & P_T(\pi) \\ 0 & E \end{pmatrix} \cdot \begin{pmatrix} P_{\bar{T}}^K(\pi) & B(\pi) \\ 0 & E \end{pmatrix} = \begin{pmatrix} P_{\bar{T}}^{K+1}(\pi) & \tilde{B}(\pi) \\ 0 & E \end{pmatrix}$$

mit

$$\tilde{B}(\pi) = P_{\bar{T}}(\pi)B(\pi) + P_T(\pi).$$

Für die Elemente von P^{K+1} gilt

$$p_{il}^{(K+1)} = \sum_{j=1}^n p_{ij} p_{jl}^{(K)}, \quad i, l = 1, \dots, n.$$

Da $p_{ij} \geq 0$ und nach Induktionsannahme auch $p_{ij}^{(K)} \geq 0$ ist, gilt $p_{ij}^{(K+1)} \geq 0$; und für die Zeilensummen:

$$\sum_{l=1}^n p_{il}^{(K+1)} = \sum_{l=1}^n \sum_{j=1}^n p_{ij} p_{jl}^{(K)} = \sum_{j=1}^n \left[p_{ij} \left(\sum_{l=1}^n p_{jl}^{(K)} \right) \right]. \quad (6.44)$$

Weil P^K nach Annahme eine stochastische Matrix ist, ist die innere Summe in (6.44) für jedes $j = 1, \dots, n$ gleich eins und somit

$$\sum_{l=1}^n p_{il}^{(K+1)} = \sum_{j=1}^n [p_{ij} \cdot 1] = 1.$$

23. $0_{k \times (n-k)}$ bezeichnet die $k \times (n-k)$ -Nullmatrix.

Damit ist gezeigt: Wenn (6.43) für K gilt, dann gilt (6.43) auch für $K + 1$ und P^{K+1} ist auch eine stochastische Matrix. Durch Induktionsschluss ist dann P^K für beliebige $K = 1, 2, \dots$ eine stochastische Matrix und hat die Gestalt (6.43). ■

Der folgende Satz beantwortet die **Frage 1** auf Seite 264.

Satz 6.6: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung (S, X, T, U, P, C, G) nach Def. 6.10. Die Politik π ist genau dann zulässig, wenn es eine natürliche Zahl $1 \leq K \leq k$ gibt, so dass für die Übergangsmatrix (6.37) gilt:

$$\|P_{\bar{T}}^K(\pi)\|_{\infty} < 1$$

wobei k die Anzahl der Zustände in \bar{T} bezeichnet²⁴. □

Beweis:

Notwendige Richtung: Sei π eine Politik, so dass $\|P_{\bar{T}}^K(\pi)\|_{\infty} < 1$ für ein $K \in \mathbb{N}$ mit $1 \leq K \leq k$. Die Matrix

$$P := (p_{ij}(\pi))_{i,j} = \begin{pmatrix} P_{\bar{T}}^K(\pi) & P_T(\pi) \\ 0 & E \end{pmatrix}$$

ist gemäß Def. 6.10 eine stochastische Matrix, das heißt $p_{ij}(\pi) \in [0, 1]$ für alle i, j und $\sum_j p_{ij}(\pi) = 1$ für jedes i . Gemäß Hilfssatz 6.5 ist

$$P^K = \begin{pmatrix} P_{\bar{T}}^K(\pi) & B(\pi) \\ 0 & E \end{pmatrix} := (p_{ij}^{(K)})_{i,j}$$

wobei $B(\pi)$ eine Funktion von $P_{\bar{T}}^K(\pi)$ und $P_T(\pi)$ ist.

Wenn $\|P_{\bar{T}}^K(\pi)\|_{\infty} < 1$ ist, dann gilt für alle $i = 1, \dots, k$: $\sum_{j=1}^k p_{ij}^{(K)}(\pi) < 1$, und weil $P^K(\pi)$ eine stochastische Matrix ist, also $\sum_{j=1}^n p_{ij}^{(K)}(\pi) = 1$ für alle $i = 1, \dots, k$, folgt

$$\sum_{j=k+1}^n p_{ij}^{(K)}(\pi) > 0 \text{ für alle } i = 1, \dots, k. \quad (6.45)$$

24. Die Matrixnorm ist die Zeilensummennorm $\|A\|_{\infty} := \max_i \sum_j |a_{ij}|$, $A = (a_{ij})$. Sie ist im $\mathbb{R}^{n \times n}$ eine Matrixalgebra norm. Daher gilt für beliebige Matrizen $A, B \in \mathbb{R}^{n \times n}$ die Abschätzung $\|AB\|_{\infty} \leq \|A\|_{\infty} \|B\|_{\infty}$ [72].

Dann ist mindestens ein Summand in (6.45) positiv, das heißt zu jedem $i = 1, \dots, k$ gibt es ein $j \in \{k+1, \dots, n\}$, für das gilt:

$$p_{ij}^{(K)}(\pi) > 0.$$

Damit ist die Politik π gemäß Def. 6.12 zulässig.

Hinreichende Richtung: Sei π zulässig, das heißt für jeden nicht beendeten Zustand $\sigma_i \in \bar{T}$ ($i = 1, \dots, k$) gibt es einen Endzustand $\sigma_{j(i)} \in T$ ($j(i) \in \{k+1, \dots, n\}$), der unter der Politik π erreichbar ist, das heißt es gibt ein $K(i)$, $1 \leq K(i) \leq k$, $k := |\bar{T}|$ und eine positive $K(i)$ -Schritt Übergangswahrscheinlichkeit:

$$p_{i,j(i)}^{(K(i))}(\pi) := p^{(K(i))}(\sigma_{j(i)} | \sigma_i, \pi(\sigma_i)) > 0. \quad (6.46)$$

Sei o. B. d. A. $i^* \in \{1, \dots, k\}$ so ausgewählt, dass für all $i = 1, \dots, k$ gilt:

$$\sum_{j=k+1}^n p_{i^*,j}^{(K(i^*))}(\pi) \leq \sum_{j=k+1}^n p_{i,j}^{(K(i^*))}(\pi).$$

Weil P eine stochastische Matrix ist, folgt daraus für all $i = 1, \dots, k$:

$$\sum_{j=1}^k p_{i^*,j}^{(K(i^*))}(\pi) \geq \sum_{j=1}^k p_{i,j}^{(K(i^*))}(\pi)$$

und damit wegen (6.46):

$$\|P_{\bar{T}}^{K(i^*)}(\pi)\|_{\infty} = \sum_{j=1}^k p_{i^*,j}^{(K(i^*))}(\pi) = 1 - \sum_{j=k+1}^n p_{i^*,j}^{(K(i^*))}(\pi) \leq 1 - p_{i^*,j(i^*)}^{(K(i^*))}(\pi) < 1.$$

Also ist π zulässig. ■

Anmerkungen zu Satz 6.6

Die beiden folgenden Anmerkungen setzen die Anmerkungen auf Seite 263 fort.

1. Der Satz besagt anschaulich das Folgende: Wenn für jeden Anfangszustand die Summe aller K -Schritt-Übergangswahrscheinlichkeiten kleiner als eins ist, dann gibt es für jeden Anfangszustand eine positive Wahrscheinlichkeit, dass in K -Schritten ein Endzustand erreicht wird.
2. Wenn für eine Politik π die Menge der Endzustände T von jedem Anfangszustand erreichbar ist, dann ist \bar{T} nicht abgeschlossen und $P(\pi)$ in (6.35) ist die kanonische Form der Übergangsmatrix der Markov-Kette.

Korollar 6.6.1: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung (S, X, T, U, P, C, G) nach Def. 6.10. Und es sei π eine

zulässige Politik. Dann ist unter der Politik π jeder Zustand aus \bar{T} transient, und aus jedem nicht beendeten Zustand in \bar{T} ist die Menge der beendeten Zustände T mit Wahrscheinlichkeit eins erreichbar.

□

Beweis: Die beiden Aussagen werden getrennt bewiesen.

Erste Aussage: Ohne Beschränkung der Allgemeinheit sei σ_i ein beliebiger Zustand aus \bar{T} . Nach [134] ist σ_i genau dann *transient*, wenn gilt:

$$E\{\rho_i\} = \sum_{\kappa=0}^{\infty} p_{ii}^{(\kappa)}(\pi) < \infty, \quad (6.47)$$

wobei ρ_i bezeichnet, wie oft, beginnend im Zustand σ_i , der Zustand σ_i wieder erreicht wird. Da $P(\pi)$ eine stochastische Matrix ist, gilt:

$$\sum_{\kappa=0}^{\infty} p_{ii}^{(\kappa)}(\pi) \leq \sum_{\kappa=0}^{\infty} \sum_{j=1}^k p_{ij}^{(\kappa)}(\pi) \leq \sum_{\kappa=0}^{\infty} \|P_{\bar{T}}^{\kappa}(\pi)\|_{\infty}. \quad (6.48)$$

Entsprechend der Annahme ist π eine zulässige Politik und deshalb gibt es nach Satz 6.6 ein K mit $1 \leq K \leq k$ und $k = |\bar{T}|$, so dass für alle $\kappa \geq K$ gilt:

$$\|P_{\bar{T}}^{\kappa}(\pi)\|_{\infty} < 1. \quad (6.49)$$

Sei nun

$$\kappa = \mu K + \nu \text{ mit } \mu, \nu \in \mathbb{N}_0 \text{ und } 0 \leq \nu \leq K-1, \quad (6.50)$$

dann ist

$$\|P_{\bar{T}}^{\kappa}(\pi)\|_{\infty} = \|P_{\bar{T}}^{\mu K + \nu}(\pi)\|_{\infty} \leq \|P_{\bar{T}}^K(\pi)\|_{\infty}^{\mu} \cdot \|P_{\bar{T}}^{\nu}(\pi)\|_{\infty} \leq \|P_{\bar{T}}^K(\pi)\|_{\infty}^{\mu} \quad (6.51)$$

und damit wegen (6.49)

$$\sum_{\kappa=0}^{\infty} \|P_{\bar{T}}^{\kappa}(\pi)\|_{\infty} = \sum_{\mu=0}^{\infty} \sum_{\nu=0}^{K-1} \|P_{\bar{T}}^{\mu K + \nu}(\pi)\|_{\infty} \leq K \sum_{\mu=0}^{\infty} \|P_{\bar{T}}^K(\pi)\|_{\infty}^{\mu} = \frac{K}{1 - \|P_{\bar{T}}^K(\pi)\|_{\infty}} < \infty.$$

Mit (6.48) folgt schließlich

$$\sum_{\kappa=0}^{\infty} p_{ii}^{(\kappa)}(\pi) < \infty,$$

womit (6.47) gezeigt ist.

Zweite Aussage: Gesucht ist die Wahrscheinlichkeit

$$W\{\exists \kappa \in \mathbb{N} : x_{t+\kappa} \in T \mid x_t \in \bar{T}\}.$$

Die Wahrscheinlichkeit, dass vom beliebigen nicht beendeten Zustand $\sigma_i \in \bar{T}$ einer der beendeten Zustände $\sigma_{k+1}, \dots, \sigma_n$ in κ Schritten erreichbar ist, beträgt

$$\sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi).$$

Damit ist

$$W\{\exists \kappa \in \mathbb{N} : x_{t+\kappa} \in T \mid x_t \in \bar{T}\} = \lim_{\kappa \rightarrow \infty} \sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi). \quad (6.52)$$

Da $P(\pi)$ eine stochastische Matrix ist, gilt für alle $\kappa = 1, 2, \dots$:

$$\sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi) = 1 - \sum_{j=1}^k p_{ij}^{(\kappa)}(\pi). \quad (6.53)$$

Entsprechend der Annahme ist π eine zulässige Politik, so dass (6.49) gilt und mit (6.50) auch (6.51). Damit folgt aus (6.53):

$$\sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi) \geq 1 - \|P_{\bar{T}}^{\kappa}(\pi)\|_{\infty}. \quad (6.54)$$

und

$$\sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi) \geq 1 - \|P_{\bar{T}}^{\mu}(\pi)\|_{\infty}^{\mu}. \quad (6.55)$$

Für $\kappa \rightarrow \infty$ gilt wegen (6.50) auch $\mu \rightarrow \infty$, und so folgt mit (6.55):

$$\lim_{\kappa \rightarrow \infty} \sum_{j=k+1}^n p_{ij}^{(\kappa)}(\pi) \geq 1 - \lim_{\mu \rightarrow \infty} \|P_{\bar{T}}^{\mu}(\pi)\|_{\infty}^{\mu} = 1. \quad (6.56)$$

Daraus ergibt sich mit (6.52):

$$W\{\exists \kappa \in \mathbb{N} : x_{t+\kappa} \in T \mid x_t \in \bar{T}\} = 1.$$

■

Anmerkung

Entscheidend in der Beweisführung ist die Tatsache, dass die Politik π zulässig ist. Diese Eigenschaft gewährleistet, dass (6.56) für jeden Ausgangszustand $\sigma_i \in \bar{T}$ gilt, so dass für x_t ein beliebiger nicht beendeter Zustand angenommen werden kann.

Die **Frage 2** auf Seite 264 formuliert ein scheinbares „Dilemma“: Einerseits steht die Anzahl der Schritte zum Erreichen eines beendeten Zustands aus einem nicht beendeten Zustand nicht im vornherein fest, da die Zustandsübergänge stochastisch sind. Und somit ist die Anzahl der mit den unscharfen Nebenbedingungen zu bewertenden Entscheidungen nicht bekannt. Andererseits gehen diese Entscheidungen in die Berechnung der unscharfen Entscheidung μ_D ein. Der „Trick“ zur Lösung des „Dilemmas“ besteht zum einen in der Einführung von Vektoren (Def. 6.11), wodurch die Berechnung von μ_D in ein Fixpunktproblem (6.41) überführt werden kann, und zum anderen in der Anwendung eines geeigneten Fixpunktsatzes. Das Ergebnis ist Satz 6.9, der zeigt, wie für eine gegebene Politik π der unscharfe Entscheidungsvektor $\mu_D(\pi)$ berechnet werden kann. Zu seinem Beweis werden ein Hilfssatz und der *Weissingersche Fixpunktsatz* benötigt, die zunächst eingeführt werden.

Hilfssatz 6.7: Für alle $a, b, c \in \mathbb{R}$ und den Minimum-Operator \wedge gilt

$$|(a \wedge b) - (a \wedge c)| \leq |b - c|. \quad (6.57)$$

□

Beweis: Ohne Beschränkung der Allgemeinheit sei $b \leq c$. Bezüglich des Wertes a sind drei Fälle zu unterscheiden:

1. $a \leq b \leq c$: $|(a \wedge b) - (a \wedge c)| = |a - a| = 0 \leq |b - c|$;
2. $b \leq a \leq c$: $|(a \wedge b) - (a \wedge c)| = |b - a| = a - b \leq c - b = |b - c|$;
3. $b \leq c \leq a$: $|(a \wedge b) - (a \wedge c)| = |b - c|$

■

Satz 6.8 (Weissingerscher Fixpunktsatz): Sei $\sum \alpha_n$ eine konvergente Reihe mit nichtnegativen Gliedern, Ω eine nichtleere abgeschlossene Teilmenge eines vollständigen metrischen Raumes mit der Metrik d und T eine Selbstabbildung von Ω mit

$$d(T^n(\omega'), T^n(\omega'')) \leq \alpha_n d(\omega', \omega'') \text{ für alle } \omega', \omega'' \in \Omega, n \in \mathbb{N}^{25}). \quad (6.58)$$

Dann besitzt T genau einen Fixpunkt $\tilde{\omega}$ in Ω , dieser Fixpunkt ist Grenzwert der Iterationsfolge $(T^n \omega^{(0)})$ bei beliebigem $\omega^{(0)} \in \Omega$, und es gilt die Fehlerabschätzung

25. T^n bezeichnet die n -fache Anwendung der Abbildung T .

$$d(\tilde{\omega}, T^n \omega^{(0)}) \leq \left(\sum_{\nu=n}^{\infty} \alpha_{\nu} \right) d(T \omega^{(0)}, \omega^{(0)}) \quad 26).$$

□

Satz 6.9: Gegeben seien ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit (S, X, T, U, P, C, G) nach Def. 6.10, eine zulässige Politik $\pi \in \Pi_p$ und die Übergangsmatrix

$$P(\pi) := (p_{ij}(\pi))_{i,j} = \begin{pmatrix} P_{\bar{T}}(\pi) & P_T(\pi) \\ 0 & E \end{pmatrix},$$

das heißt, gemäß Satz 6.6 gibt es ein $K \in \mathbb{N}$ mit $1 \leq K \leq k$, so dass $\|P_{\bar{T}}^K(\pi)\|_{\infty} < 1$.

Dann besitzt die Transformation (6.41)

$$\begin{aligned} T_{\pi} &: [0, 1]^k \rightarrow [0, 1]^k \\ T_{\pi}(\omega) &:= \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \omega + P_T(\pi) \cdot \mu_G] \end{aligned} \quad (6.59)$$

einen eindeutigen Fixpunkt $\tilde{\omega} = \mu_D(\pi)$, und die Iterationsfolge $(\omega^{(n)})$ mit $\omega^{(n+1)} = T(\omega^{(n)})$ konvergiert für beliebige Anfangswerte $\omega^{(0)}$ gegen den Fixpunkt. Dabei gilt die Fehlerabschätzung

$$\|\tilde{\omega} - T_{\pi}^n(\omega^{(0)})\|_{\infty} \leq K \frac{\|P_{\bar{T}}^n(\pi)\|_{\infty}}{1 - \|P_{\bar{T}}^K(\pi)\|_{\infty}} \cdot \|T_{\pi}(\omega^{(0)}) - \omega^{(0)}\|_{\infty}.$$

□

Beweis: Man betrachtet die Teilmenge $\Omega = [0, 1]^k$ des *Banachraumes* \mathbb{R}^k mit der *Maximumnorm* $\|x\|_{\infty} = \max_i |x_i|$ und der kanonischen Metrik $d(x, y) = \|x - y\|_{\infty}$ für alle $x, y \in \mathbb{R}^k$. Die Menge Ω ist *abgeschlossen*, was aus der Abgeschlossenheit des Intervalls $[0, 1]$ folgt. Die Maximumnorm $\|x\|_{\infty}$ für reelle Vektoren des \mathbb{R}^k ist *verträglich*²⁷⁾ mit der *Zeilensummennorm*²⁴⁾ $\|A\|_{\infty} = \max_{i=1}^k \sum_{j=1}^k |a_{ij}|$ für Matrizen $A = (a_{ij}) \in \mathbb{R}^{k \times k}$ (vgl. [72]).

26. Die hier verwendete Fassung des Fixpunktsatzes von Johannes Weissinger folgt der Formulierung in [72], wo der Satz als Übungsaufgabe gestellt wird. Sie ist klarer als die ursprüngliche Formulierung in [167], wo der Satz bewiesen wird.

27. Das heißt: $\|Ax\|_{\infty} \leq \|A\|_{\infty} \|x\|_{\infty}$. Wegen der Verträglichkeit beider Normen werden sie der Einfachheit halber mit dem gleichen Symbol bezeichnet.

Die Transformation T_π in (6.59) ist eine Selbstabbildung der Menge Ω in sich, da $\mu_C(\pi) \in [0, 1]^k$ und der Minimumsoperator \wedge komponentenweise angewendet wird.

Zur Vereinfachung wird in (6.59) $P_{\bar{T}}(\pi)$ durch $A = (a_{ij}) = (a_{ij}^{(1)})$ und $P_T(\pi) \cdot \mu_G$ durch b sowie $\mu_C(\pi)$ durch c ersetzt. Bei den Komponenten der Transformation (6.59) wird der Einfachheit halber das π weggelassen, das heißt $T_\pi(\omega) = (T_1(\omega), \dots, T_k(\omega))^T$. Sei $A = (a_{ij})$ die Übergangsmatrix $P_{\bar{T}}(\pi)$ und sei $A^n = (a_{ij}^{(n)})$ die n -Schritt-Übergangsmatrix $P_{\bar{T}}^{(n)}(\pi)$, wobei $P_{\bar{T}}^{(0)}(\pi)$ als die $k \times k$ -Einheitsmatrix definiert sei. Damit ist

$$T_\pi(\omega) = \begin{pmatrix} T_1(\omega) \\ \dots \\ T_k(\omega) \end{pmatrix} = \begin{pmatrix} c_1 \\ \dots \\ c_k \end{pmatrix} \wedge \left[\begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \cdot \begin{pmatrix} \omega_1 \\ \dots \\ \omega_k \end{pmatrix} + \begin{pmatrix} b_1 \\ \dots \\ b_k \end{pmatrix} \right].$$

Um den Weissingerschen Fixpunktsatz anwenden zu können, ist die Gültigkeit von (6.58) für die hier betrachtete Transformation T_π und die Konvergenz der entsprechenden Reihe $\sum \alpha_n$ zu zeigen.

Zunächst wird durch vollständige Induktion nach n bewiesen, dass für alle T_i , $i = 1, \dots, k$ und alle $n = 0, 1, \dots$ folgende Abschätzung gilt:

$$|T_i^n(\omega') - T_i^n(\omega'')| \leq \sum_{j=1}^k a_{ij}^{(n)} |\omega_j' - \omega_j''|. \quad (6.60)$$

Für $n = 0$ sei $T^0 := I$ die identische Abbildung $I(\omega) = \omega$ für alle $\omega \in \Omega$. Dann ist

$$|T_i^n(\omega') - T_i^n(\omega'')| = |I_i(\omega') - I_i(\omega'')| = |\omega_i' - \omega_i''| \leq \sum_{j=1}^k \delta_{ij} |\omega_j' - \omega_j''| = \sum_{j=1}^k a_{ij}^{(n)} |\omega_j' - \omega_j''|,$$

wobei δ_{ij} das Kronecker-Symbol²⁸⁾ ist.

Nun wird angenommen, dass (6.60) für alle $i = 1, \dots, k$ und ein beliebiges, aber festes $n > 0$ gelte. Dann ist

$$\begin{aligned} |T_i^{n+1}(\omega') - T_i^{n+1}(\omega'')| &= |T_i[T^n(\omega')] - T_i[T^n(\omega'')]| \\ &= \left| (c_i \wedge [\sum_{j=1}^k a_{ij} T_j^n(\omega') + b_i]) - (c_i \wedge [\sum_{j=1}^k a_{ij} T_j^n(\omega'') + b_i]) \right| \end{aligned}$$

und nach Anwendung von Hilfssatz 6.7 und der Dreiecksungleichung:

$$|T_i^{n+1}(\omega') - T_i^{n+1}(\omega'')| \leq \left| \sum_{j=1}^k a_{ij} [T_j^n(\omega') - T_j^n(\omega'')] \right| \leq \sum_{j=1}^k a_{ij} |T_j^n(\omega') - T_j^n(\omega'')|.$$

Mit der Induktionsannahme ergibt sich schließlich

28. siehe Fußnote 5 auf Seite 121

$$|T_1^{n+1}(\omega') - T_1^{n+1}(\omega'')| \leq \sum_{j=1}^k a_{ij} [\sum_{l=1}^k a_{jl}^{(n)} |\omega_l' - \omega_l''|] = \sum_{l=1}^k a_{il}^{(n+1)} |\omega_l' - \omega_l''|.$$

Daraus kann geschlossen werden, dass (6.60) für alle $n = 0, 1, \dots$ gilt.

Mit Hilfe von (6.60) lässt sich zeigen, dass für jedes n die Transformation T^n dehnungsbeschränkt im Sinne von (6.58) ist:

$$\|T^n(\omega') - T^n(\omega'')\|_\infty = \max_i |T_i^n(\omega') - T_i^n(\omega'')| \leq \max_i \sum_{j=1}^k a_{ij}^{(n)} |\omega_j' - \omega_j''|$$

und aufgrund der Verträglichkeit der Matrixnorm $\|\cdot\|_\infty$ und der Vektornorm $\|\cdot\|_\infty$ gilt mit $\tilde{\omega}_j = |\omega_j' - \omega_j''|$ (und wegen $a_{ij}^{(n)} \geq 0$):

$$\|T^n(\omega') - T^n(\omega'')\|_\infty \leq \max_i \left| \sum_{j=1}^k a_{ij}^{(n)} \tilde{\omega}_j \right| = \|A^n \tilde{\omega}\|_\infty \leq \|A^n\|_\infty \|\tilde{\omega}\|_\infty = \|A^n\|_\infty \|\omega_j' - \omega_j''\|_\infty$$

woraus schließlich mit $\alpha_n := \|A^n\|_\infty$ die Dehnungsbeschränktheit nach (6.58) folgt.

Jetzt bleibt zu zeigen, dass die Reihe $\sum_{n=0}^\infty \alpha_n$ konvergiert. Dazu wird die Reihe in Teilsommen der Länge K zerlegt, wobei $1 \leq K \leq k$, so dass gilt $\|A^K\|_\infty < 1$ und mit $k = |\bar{T}|$. Damit ist

$$\sum_{n=0}^\infty \alpha_n = \sum_{n=0}^\infty \|A^n\|_\infty = \sum_{n=0}^\infty \sum_{i=0}^{K-1} \|A^{nK+i}\|_\infty. \quad (6.61)$$

Weil A die Untermatrix einer stochastischen Matrix ist, gilt $\|A\|_\infty \leq 1$. Damit folgt, dass die Reihenglieder schwach monoton abnehmen, denn es gilt

$$\|A^{n+1}\|_\infty = \|AA^n\|_\infty \leq \|A\|_\infty \|A^n\|_\infty \leq \|A^n\|_\infty^{24}.$$

Dies erlaubt eine Abschätzung von (6.61) nach oben:

$$\sum_{n=0}^\infty \alpha_n \leq \sum_{n=0}^\infty K \|A^{nK}\|_\infty \leq \sum_{n=0}^\infty K \|A^K\|_\infty^n = K \sum_{n=0}^\infty \|P_T^K(\pi)\|_\infty^n.$$

Diese geometrische Reihe konvergiert, weil nach Voraussetzung π eine zulässige Politik ist und deshalb $\|P_T^K(\pi)\|_\infty < 1$ gilt.

Damit sind alle Voraussetzungen des Weissingerschen Fixpunktsatzes (Satz 6.8) gezeigt und es folgt, dass T genau einen Fixpunkt $\hat{\omega}$ in Ω besitzt und dieser Fixpunkt der Grenzwert der Iterationsfolge $(T^n \omega^{(0)})$ bei beliebigem $\omega^{(0)} \in \Omega$ ist. Weiterhin folgt daraus die Abschätzung

$$d(\tilde{\omega}, T_\pi^n(\omega^{(0)})) \leq \left(\sum_{\nu=n}^\infty \alpha_\nu \right) d(T_\pi(\omega^{(0)}), \omega^{(0)}) = \left(\sum_{\nu=n}^\infty \|A^\nu\|_\infty \right) d(T_\pi(\omega^{(0)}), \omega^{(0)}).$$

Somit bleibt zu zeigen, dass $\sum_{\nu=n}^\infty \|A^\nu\|_\infty = K \frac{\|A^n\|_\infty}{1 - \|A^K\|_\infty}$. Für die geometrische Reihe mit

$|q| < 1$ gilt $\sum_{\nu=0}^\infty q^\nu = \frac{1}{1-q}$. Damit ist

$$\begin{aligned} \sum_{\nu=n}^{\infty} \|A^{\nu}\|_{\infty} &\leq \|A^n\|_{\infty} \sum_{\nu=0}^{\infty} \|A^{\nu}\|_{\infty} = \|A^n\|_{\infty} \sum_{\nu=0}^{\infty} \sum_{i=0}^{K-1} \|A^{\nu K+i}\|_{\infty} \\ &\leq K \|A^n\|_{\infty} \sum_{\nu=0}^{\infty} \|A^K\|_{\infty}^{\nu} = K \frac{\|A^n\|_{\infty}}{1 - \|A^K\|_{\infty}} \end{aligned}$$

■

Die Umsetzung des Verfahrens in ein Programm ist im Abschnitt 6.6.3 beschrieben. Ein Rechenbeispiel wird im Abschnitt 6.6.4 vorgestellt.

6.4.3 Optimale Politik

Auf die **Frage 3** auf Seite 264 nach der Existenz einer optimalen Politik kann eine *existenzielle* und eine *konstruktive* Antwort gegeben werden. Die Existenz einer optimalen Politik kann aus der Struktur der Menge der zulässigen Politiken abgeleitet werden (siehe die folgenden Def. 6.13, Satz 6.16 und Satz 6.17). Eine konstruktive Antwort gibt in diesem Abschnitt der Satz 6.20, der zeigt wann eine optimale Politik existiert und wie sie berechnet werden kann. Seine Beweisführung bedient sich, wie Satz 6.9 oben, des Weissingerschen Fixpunktsatzes und ist unabhängig von der zunächst gezeigten Existenz einer optimalen Politik.

Definition 6.13: *Besser-Relation der Politiken, optimale Politik.*

Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π . Eine Politik $\pi' \in \Pi$ ist *besser* als eine Politik $\pi'' \in \Pi$ – in Zeichen $\pi' \geq \pi''$ – genau dann, wenn für die zugehörigen Entscheidungsvektoren $\mu_D(\pi')$ und $\mu_D(\pi'')$ (vgl. Def. 6.11) gilt:

$$\forall i = 1, \dots, k : \mu_D(\pi', \sigma_i) \geq \mu_D(\pi'', \sigma_i) \quad ^{29}). \quad (6.62)$$

Eine Politik aus Π wird *optimal*, π^{opt} genannt, genau dann, wenn sie zulässig ist, $\pi^{\text{opt}} \in \Pi_p$, und wenn $\pi^{\text{opt}} \geq \pi$ für jede Politik $\pi \in \Pi_p$ gilt. Dann kann auch

$\pi^{\text{opt}} = \max_{\pi \in \Pi_p} \pi$ geschrieben werden. □

Anmerkung

Diese Definition ist eine Konkretisierung von Def. 6.8 für Entscheidungsprozesse (S, X, T, U, P, C, G) .

29. Für die Definition reicht es, die Zustände $\sigma_1, \dots, \sigma_k$ zu betrachten, weil die unscharfen Entscheidungen für die terminierten Zustände mit dem unscharfen Ziel übereinstimmen, $\mu_D(\sigma_i) = \mu_G(\sigma_i)$ für $i = k+1, \dots, n$, also von Politiken unabhängig sind.

Hilfssatz 6.10: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π . Dann bildet Π mit der Besser-Relation aus Def. 6.13, (Π, \geq) eine *Quasi-Ordnung*.

□

Beweis: Es ist zu zeigen, dass die Besser-Relation \geq auf der Menge Π reflexiv und transitiv ist [23]. Da die Größer-Gleich-Relation auf \mathbb{R} reflexiv und transitiv ist, folgen diese Eigenschaften für die Besser-Relation zwischen Politiken unmittelbar aus Def. 6.13.

■

Anmerkungen zu Hilfssatz 6.10

1. Eine optimale Politik muss nicht existieren, denn da Π_p durch die *Besser-Relation* nur quasi-geordnet ist, ist es möglich, dass es $\pi^a, \pi^b \in \Pi_p$ mit folgenden Eigenschaften gibt: $\pi^a \not\geq \pi^b$ und $\pi^b \not\geq \pi^a$, aber $\pi^a \geq \pi$ für alle $\pi \in \Pi_p \setminus \{\pi^b\}$ und $\pi^b \geq \pi$ für alle $\pi \in \Pi_p \setminus \{\pi^a\}$, und es gibt kein $\pi \in \Pi \setminus \{\pi^a, \pi^b\}$ mit $\pi \geq \pi^a, \pi^b$ ³⁰⁾.
2. Wenn es in Π_p eine optimale Politik gibt, dann ist sie nicht notwendigerweise eindeutig. Angenommen, es gäbe zwei optimale Politiken $\pi', \pi'' \in \Pi_p$. Dann würde sowohl $\pi' \geq \pi''$ als auch $\pi'' \geq \pi'$ gelten, also:

$$\forall i = 1, \dots, k : [\mu_D(\pi', \sigma_i) \geq \mu_D(\pi'', \sigma_i)] \wedge [\mu_D(\pi'', \sigma_i) \geq \mu_D(\pi', \sigma_i)].$$

Somit würden die Entscheidungsvektoren der beiden Politiken übereinstimmen. Die Politiken selbst, müssen jedoch nicht übereinstimmen.

3. Aus Anmerkung 2 wird deutlich, dass die Besser-Relation auf Π *nicht antisymmetrisch* ist. Und deshalb ist auch (Π, \geq) *keine Halbordnung*.

Für die weitere Untersuchung ist eine Halbordnungseigenschaft zwischen Politiken erwünscht. Da sie auf der Menge Π nicht möglich ist, wird eine „geeignete Menge“ mit einer Halbordnung gesucht. Der folgende Hilfssatz aus [23] erweist sich dabei als nützlich. Das Endresultat ist Satz 6.14.

Hilfssatz 6.11: In einer beliebigen quasi-geordneten Menge $(M, >)$ sei eine Relation \sim wie folgt definiert: Für alle $a, b \in M$ gilt $a \sim b$ genau dann, wenn $a \geq b$ und $b \geq a$. Dann gelten folgende Aussagen:

1. Die Relation \sim ist eine *Äquivalenzrelation* auf M .

30. Beispiel: $\mu_D(\pi^a) = (1, 7/10, 4/10)^T$, $\mu_D(\pi^b) = (1, 4/10, 7/10)^T$ und für alle anderen $\pi \in \Pi$ gilt $\mu_D(\pi) \in [0, 1] \times [0, 4/10] \times [0, 4/10]$.

2. Wenn $[x]$ und $[y]$ zwei Äquivalenzklassen bezüglich der Relation \sim sind, dann gilt $x \geq y$ entweder für kein $a \in [x]$, $b \in [y]$ oder für alle $a \in [x]$, $b \in [y]$.
3. Die Faktormenge³¹⁾ M/\sim bildet mit einer Relation \geq eine *Halbordnung*, wenn die Relation so definiert ist, dass $[x] \geq [y]$ bedeutet: $a \geq b$ für einige (und damit alle) $a \in [x]$ und $b \in [y]$.

□

Beweis: siehe [23].

Definition 6.14: *Ähnlichkeit von Politiken.*

Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π und der *Quasi-Ordnung* (Π, \geq) gemäß Hilfssatz 6.10. In Π sei die Ähnlichkeit zweier Politiken $\pi', \pi'' \in \Pi$, in Zeichen: $\pi' \sim \pi''$ wie folgt definiert:

$$(\pi' \sim \pi'') \Leftrightarrow (\pi' \geq \pi'') \wedge (\pi'' \geq \pi'). \quad (6.63)$$

□

Hilfssatz 6.12: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π und der *Quasi-Ordnung* (Π, \geq) gemäß Hilfssatz 6.10 sowie der Ähnlichkeitsrelation \sim nach Def. 6.14.

Dann ist \sim eine *Äquivalenzrelation* auf Π .

□

Beweis: Der Satz folgt unmittelbar aus Hilfssatz 6.11.

■

Definition 6.15: *Politik-Äquivalenzklasse, Politik-Faktormenge, Besser-Relation der Politik-Äquivalenzklassen.*

Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π und der *Quasi-Ordnung* (Π, \geq) gemäß Hilfssatz 6.10 und der Äquivalenzrelation \sim nach Def. 6.14.

Die vermöge der Äquivalenzrelation \sim gebildeten Äquivalenzklassen werden *Politik-Äquivalenzklassen* genannt. Sie werden mit $[\pi]$ bezeichnet, wobei $\pi \in \Pi$ ein

31. In der Literatur ist auch die Bezeichnung *Quotientenmenge* gebräuchlich.

Repräsentant der Äquivalenzklasse ist. Die Menge der Äquivalenzklassen ist die *Politik-Faktormenge*, die mit Π/\sim bezeichnet wird.

Eine Politik-Äquivalenzklasse $[\pi'] \in \Pi/\sim$ ist *besser* als eine Politik-Äquivalenzklasse $[\pi''] \in \Pi/\sim$, in Zeichen: $[\pi'] \geq [\pi'']$, genau dann, wenn es $\pi_1 \in [\pi']$ und $\pi_2 \in [\pi'']$ gibt, so dass $\pi_1 \geq \pi_2$ gilt. \square

Anmerkung

In dieser Definition und in Def. 6.13 bezeichnet das Zeichen \geq die Größer-Gleich-Relation zwischen natürlichen Zahlen (Glg. 6.62), die Besser-Relation zwischen Politiken und die Besser-Relation zwischen Politik-Äquivalenzklassen. Weil dies unterschiedliche Relationen sind, wäre genau genommen für jede Relation ein anderes Symbol nötig. Zur Hervorhebung des Zusammenhangs zwischen diesen Relationen, der Einfachheit halber und weil sich die Bedeutung aus dem Kontext eindeutig ergibt, wird nur das Zeichen \geq verwendet. Der folgende Satz verdeutlicht dies.

Satz 6.13: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π und der Politik-Faktormenge Π/\sim gemäß Def. 6.15. Dann gelten folgende Aussagen:

1. Für beliebige $\pi', \pi'' \in [\pi]$ ist $\mu_D(\pi') = \mu_D(\pi'')$.
2. Die folgenden Aussagen sind äquivalent:
 - a) $[\pi_1] \geq [\pi_2]$;
 - b) $\pi' \geq \pi''$ für beliebige $\pi' \in [\pi_1]$ und $\pi'' \in [\pi_2]$;
 - c) $\mu_D(\pi') \geq \mu_D(\pi'')$ für beliebige $\pi' \in [\pi_1]$ und $\pi'' \in [\pi_2]$.

\square

Beweis: Zunächst wird die erste Aussage bewiesen. Wenn beliebige Politiken $\pi', \pi'' \in \Pi$ derselben Politik-Äquivalenzklassen angehören, dann sind diese Politiken äquivalent: $\pi' \sim \pi''$. Laut (6.63) in Def. 6.14 bedeutet dies, dass sowohl $\pi' \geq \pi''$ als auch $\pi'' \geq \pi'$ zutreffen. Aus Def. 6.13 folgt mit Glg. 6.62, dass zugleich $\mu_D(\pi') \geq \mu_D(\pi'')$ und $\mu_D(\pi'') \geq \mu_D(\pi')$ gelten und somit die Behauptung.

Jetzt wird bewiesen, dass aus Aussage a) die Aussage b) und aus Aussage b) die Aussage c) folgt: Wenn $[\pi_1] \geq [\pi_2]$, dann gibt es laut Def. 6.15 Politiken $\pi_1 \in [\pi_1]$ und $\pi_2 \in [\pi_2]$, so dass $\pi_1 \geq \pi_2$ gilt. Alle Politiken $\pi_1' \in [\pi_1]$ sind zu π_1 und alle Politiken $\pi_2' \in [\pi_2]$ sind zu π_2 äquivalent. Damit folgt mit Def. 6.14 und wegen der Transitivität der Besser-Relation für

Politiken $\pi_1' \geq \pi' \geq \pi''$ und $\pi' \geq \pi'' \geq \pi_2'$ und somit $\pi_1' \geq \pi_2'$. Schließlich folgt aus Def. 6.13 $\mu_D(\pi_1') \geq \mu_D(\pi_2')$.

Nun wird noch gezeigt, dass aus Aussage c) die Aussage a) folgt: Sei $\mu_D(\pi') \geq \mu_D(\pi'')$ für beliebige $\pi' \in [\pi_1]$ und $\pi'' \in [\pi_2]$. Dann gilt gemäß Def. 6.13 für beliebige $\pi' \in [\pi_1]$ und $\pi'' \in [\pi_2]$: $\pi' \geq \pi''$. Und gemäß Def. 6.15 gilt $[\pi_1] \geq [\pi_2]$. Damit ist alles gezeigt. ■

Eine Konsequenz der Aussage 1 von Satz 6.13 ist die folgende Definition.

Definition 6.16: *Entscheidungsvektor einer Politik-Äquivalenzklasse.*

Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Politik-Faktormenge Π/\sim gemäß Def. 6.15. Der *Entscheidungsvektor der Politik-Äquivalenzklasse* $[\pi] \in \Pi/\sim$ ist $\mu_D([\pi]) := \mu_D(\pi')$ für $\pi' \in [\pi]$ beliebig. □

Schließlich ist es möglich, eine Halbordnung festzulegen.

Satz 6.14: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Politik-Faktormenge Π/\sim und der Besser-Relation \geq gemäß Def. 6.15. Dann ist $(\Pi/\sim, \geq)$ eine *Halbordnung*. □

Beweis: Der Satz folgt unmittelbar aus Hilfssatz 6.11. ■

Anmerkung

In der Halbordnung $(\Pi_p/\sim, \geq)$ muss *keine* optimale Politik existieren. Da Π_p durch die *Besser-Relation* für Politik-Äquivalenzklassen nur halbgeordnet ist, ist es möglich, dass es $[\pi^a], [\pi^b] \in \Pi_p/\sim$ mit folgenden Eigenschaften gibt: $[\pi^a] \not\geq [\pi^b]$ und $[\pi^b] \not\geq [\pi^a]$, aber $[\pi^a] \geq [\pi]$ für alle $[\pi] \in (\Pi_p/\sim) \setminus \{[\pi^b]\}$ und $[\pi^b] \geq [\pi]$ für alle $[\pi] \in (\Pi_p/\sim) \setminus \{[\pi^a]\}$, und es gibt kein $[\pi] \in (\Pi_p/\sim) \setminus \{[\pi^a], [\pi^b]\}$ mit $[\pi] \geq [\pi^a], [\pi^b]$ ³².

Korollar 6.14.1: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit,

32. Beispiel: $\mu_D([\pi^a]) = (1, 7/10, 4/10)^T$, $\mu_D([\pi^b]) = (1, 4/10, 7/10)^T$ und für alle anderen $[\pi] \in \Pi_p/\sim$ gilt $\mu_D([\pi]) \in [0, 1] \times [0, 4/10] \times [0, 4/10]$. Vgl. auch Fußnote 30 auf Seite 276.

(S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Halbordnung $(\Pi_p / \sim, \geq)$ aus Satz 6.14 über der Menge zulässiger Politiken Π_p . Wenn es darin eine optimale Politik gibt, dann ist ihre *Politik-Äquivalenzklasse* eindeutig.

□

Beweis: Angenommen, es gäbe zwei optimale Politiken $\pi', \pi'' \in \Pi_p$. Dann würde sowohl $\pi' \geq \pi''$ als auch $\pi'' \geq \pi'$ gelten, also:

$$\forall i = 1, \dots, k : [\mu_D(\pi', \sigma_i) \geq \mu_D(\pi'', \sigma_i)] \wedge [\mu_D(\pi'', \sigma_i) \geq \mu_D(\pi', \sigma_i)].$$

Somit würde für die Entscheidungsvektoren der beiden Politiken $\mu_D(\pi') = \mu_D(\pi'')$ gelten. Dann wären die Politiken π', π'' gemäß Hilfssatz 6.12 äquivalent, das heißt: $[\pi'] = [\pi'']$. ■

Mit Hilfe des folgenden Hilfssatzes können zu zwei gegebenen Politiken unter bestimmten Voraussetzungen eine bessere und eine schlechtere Politik berechnet werden. Dies wird im Beweis von Satz 6.16, aus dem die Existenz einer optimalen Politik folgt, genutzt sowie für die praktische Berechnung einer optimalen Politik, deren Beschreibung im Abschnitt 6.5 folgt.

Hilfssatz 6.15: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10, mit der Menge der Politiken Π und ihrer Teilmenge der zulässigen Politiken Π_p sowie der Transformation (6.41),

$$\begin{aligned} T_\pi &: [0, 1]^k \rightarrow [0, 1]^k \\ T_\pi(\omega) &:= \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \omega + P_T(\pi) \cdot \mu_G]. \end{aligned}$$

Dann gilt:

1. Für jede Politik $\tilde{\pi} \in \Pi_p$ sind durch

$$T_{\pi_g}(\mu_D(\tilde{\pi})) := \max_{\pi \in \Pi} T_\pi(\mu_D(\tilde{\pi})) \quad (6.64)$$

und

$$T_{\pi_k}(\mu_D(\tilde{\pi})) := \min_{\pi \in \Pi} T_\pi(\mu_D(\tilde{\pi})) \quad (6.65)$$

die Politiken $\pi_g, \pi_k \in \Pi$ gegeben³³).

2. Gilt $\pi_g, \pi_k \in \Pi_p$, dann ist $\pi_g \geq \tilde{\pi}$ und $\pi_k \leq \tilde{\pi}$.

□

33. Maximum und Minimum seien hier, wie an anderen Stellen dieser Arbeit auch, komponentenweise definiert.

Beweis: Für $\tilde{\pi} \in \Pi_p$ existiert $\mu_D(\tilde{\pi}) \in [0, 1]^k$ gemäß Satz 6.9. Da Π endlich ist, gibt es ein $\pi_g \in \Pi$, so dass für $i = 1, \dots, k$ gilt:

$$\begin{aligned} & \mu_C(\pi_g(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi_g(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi_g(\sigma_i)) \cdot \mu_G] \right] \\ &= \max_{\pi \in \Pi} \left\{ \mu_C(\pi(\sigma_i) | \sigma_i) \right. \\ & \quad \left. \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{G,i}] \right] \right\}. \end{aligned}$$

Und es gibt ein $\pi_k \in \Pi$, so dass für $i = 1, \dots, k$ gilt:

$$\begin{aligned} & \mu_C(\pi_k(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi_k(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi_k(\sigma_i)) \cdot \mu_G] \right] \\ &= \min_{\pi \in \Pi} \mu_C(\pi(\sigma_i) | \sigma_i) \\ & \quad \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{G,i}] \right] \end{aligned}$$

oder – in Operatorenschreibweise – (6.64) bzw. (6.65), womit Aussage 1 gezeigt ist.

Für $i = 1, \dots, k$ gilt

$$\begin{aligned} & \max_{\pi \in \Pi} \left\{ \mu_C(\pi(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{G,i}] \right] \right\} \\ & \geq \mu_C(\tilde{\pi}(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \tilde{\pi}(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \tilde{\pi}(\sigma_i)) \cdot \mu_{G,i}] \right] \\ &= \mu_D(\tilde{\pi}). \end{aligned}$$

Genau dann ist aber

$$T_{\pi_g}(\mu_D(\tilde{\pi})) \geq \mu_D(\tilde{\pi}). \quad (6.66)$$

Entsprechend gilt für $i = 1, \dots, k$:

$$\begin{aligned} & \min_{\pi \in \Pi} \mu_C(\pi(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \pi(\sigma_i)) \cdot \mu_{G,i}] \right] \\ & \leq \mu_C(\tilde{\pi}(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} [p(\sigma_j | \sigma_i, \tilde{\pi}(\sigma_i)) \cdot \mu_{D,i}(\tilde{\pi})] + \sum_{\sigma_j \in T} [p(\sigma_j | \sigma_i, \tilde{\pi}(\sigma_i)) \cdot \mu_{G,i}] \right] \\ &= \mu_D(\tilde{\pi}). \end{aligned}$$

Das ist genau dann der Fall, wenn

$$T_{\pi_k}(\mu_D(\tilde{\pi})) \leq \mu_D(\tilde{\pi}). \quad (6.67)$$

Durch mehrfache Anwendung des Operators T_{π_g} auf beide Seiten von (6.66) bzw. des Operators T_{π_k} auf beide Seiten von (6.67) folgt

$$T_{\pi_g}^n(\mu_D(\tilde{\pi})) \geq T_{\pi_g}^{n-1}(\mu_D(\tilde{\pi})) \geq \dots \geq \mu_D(\tilde{\pi}) \text{ für } n \in \mathbb{N}.$$

bzw.

$$T_{\pi_k}^n(\mu_D(\tilde{\pi})) \leq T_{\pi_k}^{n-1}(\mu_D(\tilde{\pi})) \leq \dots \leq \mu_D(\tilde{\pi}) \text{ für } n \in \mathbb{N}.$$

Sind π_g und π_k zulässig, das heißt $\pi_g, \pi_k \in \Pi_p$, dann gilt gemäß Satz 6.9

$$T_{\pi_g}^n(\mu_D(\tilde{\pi})) \rightarrow \mu_D(\pi_g) \text{ bzw. } T_{\pi_k}^n(\mu_D(\tilde{\pi})) \rightarrow \mu_D(\pi_k) \text{ für } n \rightarrow \infty$$

und damit $\mu_D(\pi_g) \geq \mu_D(\tilde{\pi})$ bzw. $\mu_D(\pi_k) \leq \mu_D(\tilde{\pi})$, also mit Def. 6.13

$$\pi_g \geq \tilde{\pi} \text{ bzw. } \pi_k \leq \tilde{\pi},$$

womit Aussage 2 gezeigt ist. ■

Satz 6.16: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Halbordnung $(\Pi/\sim, \geq)$ aus Satz 6.14.

Wenn alle Politiken zulässig sind, das heißt $\Pi = \Pi_p$, dann ist die Politik-Faktormenge Π/\sim ein *endlicher vollständiger Verband*. □

Beweis: Es wird gezeigt, dass die Politik-Faktormenge Π/\sim *endlich* viele Elemente enthält, dass die Halbordnung $(\Pi/\sim, \geq)$ die algebraische Struktur eines *Verbands* besitzt, und dass dieser *vollständig* ist.

Endlichkeit: Eine Politik π aus der Menge aller Politiken Π ist durch den Vektor $\pi = (\pi(\sigma_1), \dots, \pi(\sigma_k))^T$ gegeben, der so viele Komponenten hat, wie die Menge der nicht-beendeten Zustände Elemente besitzt. Für jeden Zustand σ_i , $i = 1, \dots, k$ gilt: $\pi(\sigma_i) \in \{\alpha_1, \dots, \alpha_m\}$. Daraus folgt, dass Π genau $r = m^k$ unterschiedliche Politiken enthält:

$$\Pi = \{\pi^1, \dots, \pi^r\}. \quad (6.68)$$

Deshalb ist Π und damit auch Π/\sim endlich.

Verband: Die Halbordnung $(\Pi/\sim, \geq)$ ist genau dann ein Verband, wenn für zwei Elemente $[\pi'], [\pi''] \in \Pi/\sim$ das Infimum $\inf\{[\pi'], [\pi'']\}$ und das Supremum $\sup\{[\pi'], [\pi'']\}$ existieren³⁴). Unter der Annahme $\pi_k, \pi_g \in \Pi$, so dass

$$T_{\pi_k}(\mu_D^{\min}) = \min_{\pi \in \Pi} \left\{ \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \mu_D^{\min} + P_T(\pi) \cdot \mu_G] \right\}$$

bzw.

$$T_{\pi_g}(\mu_D^{\max}) = \max_{\pi \in \Pi} \left\{ \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \mu_D^{\max} + P_T(\pi) \cdot \mu_G] \right\}$$

mit

$$\mu_D^{\min} = \begin{cases} \mu_D(\pi'', \sigma_i) & \text{für } \mu_D(\pi', \sigma_i) \geq \mu_D(\pi'', \sigma_i) \\ \mu_D(\pi', \sigma_i) & \text{für } \mu_D(\pi', \sigma_i) < \mu_D(\pi'', \sigma_i) \end{cases} \quad (6.69)$$

bzw.

$$\mu_D^{\max} = \begin{cases} \mu_D(\pi'', \sigma_i) & \text{für } \mu_D(\pi', \sigma_i) < \mu_D(\pi'', \sigma_i) \\ \mu_D(\pi', \sigma_i) & \text{für } \mu_D(\pi', \sigma_i) \geq \mu_D(\pi'', \sigma_i) \end{cases}, \quad (6.70)$$

gilt dann mit Hilfssatz 6.15 $\mu_D(\pi_k) \leq \mu_D^{\min}$ bzw. $\mu_D(\pi_g) \geq \mu_D^{\max}$. Daraus folgt mit (6.69) bzw. (6.70)

$$\mu_D(\pi_k) \leq \mu_D(\pi'), \mu_D(\pi'')$$

bzw.

$$\mu_D(\pi_g) \geq \mu_D(\pi'), \mu_D(\pi'').$$

Damit ist gezeigt, dass $\pi_k \in \Pi$ eine untere Schranke und $\pi_g \in \Pi$ eine obere Schranke von π' und π'' sind. Weil Π endlich ist, folgt, dass auch eine größte untere Schranke und eine kleinste obere Schranke existieren, das heißt $\pi_k \leq \inf\{\pi', \pi''\}$ und $\pi_g \geq \sup\{\pi', \pi''\}$. Diese Eigenschaft überträgt sich gemäß Satz 6.13 von Π auf Π/\sim , das heißt $[\pi_k] \leq \inf\{[\pi'], [\pi'']\}$ und $[\pi_g] \geq \sup\{[\pi'], [\pi'']\}$.

Vollständigkeit. Ein Verband ist vollständig, wenn jede seiner Teilmengen ein Infimum und ein Supremum besitzt. Da Π/\sim endlich ist, ist diese Eigenschaft nur für endliche Mengen zu zeigen, was durch vollständige Induktion, ausgehend von zweielementigen Teilmengen leicht möglich ist (siehe z. B. [70] oder [96]). ■

34. Diese Aussage wird in der Literatur entweder als Satz [70] oder als Definition eines Verbandes [96] dargestellt.

Anmerkungen

1. In der Beweisführung dieses Satzes kommt das *Wechselprinzip* („alternation principle“) [176] zur Anwendung. Es wird von Zadeh sowohl für die Berechnung der optimalen Politik scharfer [44] als auch unscharfer Systeme [11] herangezogen. Hier reicht es jedoch nicht aus, weil μ_D^{\min} bzw. μ_D^{\max} nicht notwendigerweise unscharfe Entscheidungsvektoren von Politiken aus Π sind. Diese Politiken werden mit Hilfe von Hilfssatz 6.15 nach oben bzw. nach unten abgeschätzt.
2. Die Diskussion der Voraussetzung $\Pi = \Pi_p$ wird im Abschnitt 6.5.3 geführt.

Satz 6.17: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Halbordnung $(\Pi/\sim, \geq)$ aus Satz 6.14.

Wenn die Menge der Politiken Π nur zulässige Politiken enthält, das heißt $\Pi = \Pi_p$, dann gibt es eine eindeutige Politik-Äquivalenzklasse $[\tilde{\pi}]$, die nur optimale Politiken enthält: $\tilde{\pi} \sim \pi^{\text{opt}}$. Sie wird *optimale Politik-Äquivalenzklasse* genannt. \square

Beweis: Zunächst wird die *Existenz* einer Politik-Äquivalenzklasse, die nur optimale Politiken enthält, gezeigt: Laut Hilfssatz 6.16 ist $(\Pi/\sim, \geq)$ für $\Pi = \Pi_p$ ein vollständiger Verband, so dass alle Teilmengen von Π/\sim und damit auch die Menge Π/\sim selbst ein Infimum und ein Supremum besitzen. Für $[\tilde{\pi}] = \sup(\Pi/\sim)$ gilt $[\tilde{\pi}] \geq [\pi]$ für alle $[\pi] \in \Pi/\sim$. Dann ist gemäß Satz 6.13 $\tilde{\pi} \geq \pi$ für alle $\pi \in \Pi$ und damit $\tilde{\pi} \sim \pi^{\text{opt}}$.

Jetzt ist noch die *Eindeutigkeit* zu zeigen: Angenommen es gäbe zwei Politik-Äquivalenzklassen $[\tilde{\pi}_1]$ und $[\tilde{\pi}_2]$ mit $\tilde{\pi}_1 \sim \pi_1^{\text{opt}}$ und $\tilde{\pi}_2 \sim \pi_2^{\text{opt}}$. Dann gilt gemäß Def. 6.13 sowohl $\pi_1^{\text{opt}} \geq \pi_2^{\text{opt}}$ als auch $\pi_2^{\text{opt}} \geq \pi_1^{\text{opt}}$ und damit gemäß Def. 6.14 $\pi_1^{\text{opt}} \sim \pi_2^{\text{opt}}$, also auch $\tilde{\pi}_1 \sim \tilde{\pi}_2$, woraus die Behauptung folgt. \blacksquare

Anmerkungen zu Satz 6.17

1. In einer Politik-Äquivalenzklasse sind entweder alle Politiken optimal oder keine. Dies folgt unmittelbar aus (6.63) in Def. 6.14.
2. Die Politiken einer optimalen Politik-Äquivalenzklasse können voneinander verschieden sein; vgl. Anmerkung 2 zu Hilfssatz 6.10.
3. Die Politiken einer optimalen Politik-Äquivalenzklasse besitzen alle denselben Entscheidungsvektor; vgl. Aussage 1 in Satz 6.13.

Mit Satz 6.17 ist die Existenz einer optimalen Politik bewiesen, vorausgesetzt, die Politikmenge des betrachteten mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit enthält nur zulässige Politiken. Auf die praktische Bedeutung dieser Einschränkung wird weiter unten im Abschnitt 6.5.3 eingegangen. Ihre Ursache liegt in dem Nebeneinander von Politiken, das heißt Abbildungen aus der

Zustandsmenge in die Entscheidungsmenge, und ihren unscharfen Entscheidungsvektoren. Letztere können für eine gegebene Politik nur dann iterativ berechnet werden, wenn die Politik zulässig ist. Und umgekehrt existiert nicht zu jedem unscharfen Entscheidungsvektor eine Politik, und wenn, dann kann sie nicht ohne Weiteres berechnet werden.

Nach dieser existentiellen Antwort auf Frage 3 auf Seite 264 wird im folgenden eine konstruktive Antwort gegeben, die auch die praktische Berechnung einer optimalen Politik ermöglicht. Darauf wird aber explizit erst im Abschnitt 6.5 eingegangen.

Aus (6.40) folgt für eine optimale Politik die folgende Gleichung:

$$\mu_D^{\text{opt}} = \max_{\pi} \{ \mu_C(\pi) \wedge [P_{\bar{T}}(\pi) \cdot \mu_D^{\text{opt}} + P_T(\pi) \cdot \mu_G] \} \quad (6.71)$$

mit

$$\mu_D^{\text{opt}} := \mu_D(\pi^{\text{opt}}) = \max_{\pi} \mu_D(\pi) . \quad (6.72)$$

Wenn „max“ durch \vee ersetzt wird, dann folgt mit (6.68)

$$\begin{aligned} \mu_D^{\text{opt}} = & \{ \mu_C(\pi^1) \wedge [P_{\bar{T}}(\pi^1) \cdot \mu_D^{\text{opt}} + P_T(\pi^1) \cdot \mu_G] \vee \dots \\ & \dots \vee \mu_C(\pi^r) \wedge [P_{\bar{T}}(\pi^r) \cdot \mu_D^{\text{opt}} + P_T(\pi^r) \cdot \mu_G] \} \end{aligned}$$

oder

$$\mu_D^{\text{opt}} = \bigvee_{\nu=1}^r \{ \mu_C(\pi^{\nu}) \wedge [P_{\bar{T}}(\pi^{\nu}) \cdot \mu_D^{\text{opt}} + P_T(\pi^{\nu}) \cdot \mu_G] \} , \quad (6.73)$$

wobei \vee komponentenweise definiert ist. In (6.73) ist erkennbar, dass μ_D^{opt} ein Fixpunkt der Transformation

$$\begin{aligned} T : [0, 1]^k &\rightarrow [0, 1]^k \\ T(\omega) &= \bigvee_{\nu=1}^r \{ \mu_C(\pi^{\nu}) \wedge [P_{\bar{T}}(\pi^{\nu}) \cdot \omega + P_T(\pi^{\nu}) \cdot \mu_G] \} \end{aligned} \quad (6.74)$$

ist.

Anmerkung

Mit der Maximumbildung in (6.74) sind t-Norm und zugehörige s-Norm (vgl. Def. 2.7 und Satz 2.1) in einer Gleichung vereint. Während die vorkommende Minimumbildung tatsächlich als t-Norm gebraucht wird (vgl. Def. 6.6) rührt die Maximumbildung vom Optimierungsproblem her. Deshalb bleibt die Maximumbildung auch bestehen, wenn in Abschnitt 6.4.4 Alternativen zur t-Norm Minimumbildung untersucht werden.

Die **Frage 4** auf Seite 264 nach der Berechenbarkeit einer optimalen Politik beantwortet der Satz 6.20. Zuvor sind noch zwei Hilfssätze nötig, auf die im Beweis von Satz 6.20 zurückgegriffen wird.

Hilfssatz 6.18: Für $a, b \in \mathbb{R}$ bezeichne $a \vee b$ das Maximum zweier Elemente. Dann gilt für beliebige $a_\nu, b_\nu \in \mathbb{R}$ und $r \in \mathbb{N}$ die Abschätzung:

$$\left| \bigvee_{\nu=1}^r a_\nu - \bigvee_{\nu=1}^r b_\nu \right| \leq \bigvee_{\nu=1}^r |a_\nu - b_\nu|. \quad (6.75)$$

□

Beweis: Sei $a_\kappa = \bigvee_{\nu=1}^r a_\nu$ und $b_\lambda = \bigvee_{\nu=1}^r b_\nu$ und o. B. d. A. werde angenommen, dass $a_\kappa \geq b_\lambda$ ist. Dann gilt wegen der Monotonie der Maximumbildung $|a_\kappa - b_\lambda| \leq |a_\kappa - b_\kappa|$ und damit folgt

$$\left| \bigvee_{\nu=1}^r a_\nu - \bigvee_{\nu=1}^r b_\nu \right| \leq |a_\kappa - b_\lambda| \leq |a_\kappa - b_\kappa| \leq \bigvee_{\nu=1}^r |a_\nu - b_\nu| \quad \blacksquare$$

Hilfssatz 6.19: Für beliebige $a_\nu \in \mathbb{R}_0^+$ und $n, r \in \mathbb{N}$ gilt die Abschätzung:

$$\bigvee_{\nu=1}^r a_\nu^n \leq \left(\bigvee_{\nu=1}^r a_\nu \right)^n \quad (6.76)$$

□

Beweis: O. b. d. A. sei $a_\kappa^n := \bigvee_{\nu=1}^r a_\nu^n$. Dann gilt $\bigvee_{\nu=1}^r a_\nu^n = a_\kappa^n \leq \left(\bigvee_{\nu=1}^r a_\nu \right)^n$. ■

Berechnung der optimalen Politik

Satz 6.20: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π .

Wenn die Menge der Politiken Π nur zulässige Politiken enthält, das heißt $\Pi = \Pi_p$ bzw. für alle $\pi \in \Pi$ gibt es ein $1 \leq K \leq k$ mit $\|P_T^K(\pi)\|_\infty < 1$ ³⁵⁾, dann gilt das Folgende:

1. Das Fixpunktproblem $T(\omega) = \omega$, mit der Transformation T aus (6.74) besitzt genau eine Lösung, die durch die Iteration $\omega^{(k+1)} = T(\omega^{(k)})$ für beliebiges $\omega^{(0)} \in [0, 1]^k$ berechnet werden kann.
2. Die Lösung des Fixpunktproblems ist der eindeutige unscharfe Entscheidungsvektor einer optimalen Politik $\mu_D(\pi^{\text{opt}})$.
3. Es gilt die Fehlerabschätzung

35. vgl. Def. 6.12 und Satz 6.6

$$\|\tilde{\omega} - T^n(\omega^{(0)})\|_{\infty} \leq K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}} \cdot \|T(\omega^{(0)}) - \omega^{(0)}\|_{\infty},$$

wobei $K := \max_{\mu=1, \dots, m} K(\mu)$ mit $1 \leq K(\mu) \leq k$, $k = |\bar{T}|$, so dass $\|A^{K(\mu)}(\mu)\|_{\infty} < 1$ für $A(\mu) = P_{\bar{T}}(\pi)$ mit $p(\sigma_j | \sigma_i, a_{\mu}) = p(\sigma_j | \sigma_i, \pi(\sigma_i))$.

□

Beweis: Man betrachtet die Teilmenge $\Omega = [0, 1]^k$ des *Banachraumes*³⁶⁾ \mathbb{R}^k mit der *Maximumnorm* $\|x\|_{\infty} = \max_i |x_i|$ und der kanonischen Metrik $d(x, y) = \|x - y\|_{\infty}$ für alle $x, y \in \mathbb{R}^k$. Die Menge Ω ist *abgeschlossen*, was aus der Abgeschlossenheit des Intervalls $[0, 1]$ folgt. Die Maximumnorm $\|x\|_{\infty}$ für reelle Vektoren des \mathbb{R}^k ist *verträglich* mit der

Zeilensummennorm $\|A\|_{\infty} = \max_{i=1}^k \sum_{j=1}^k |a_{ij}|$ für Matrizen $A = (a_{ij}) \in \mathbb{R}^{k \times k}$ (vgl. Fußnote 27 auf Seite 272 und [72]). Die Transformation T in (6.74) ist eine Selbstabbildung der Menge Ω in sich, da $\mu_C(\pi^{\nu}) \in [0, 1]^k$ und der Minimumsoperator \wedge komponentenweise angewendet wird. Damit stehen in der geschweiften Klammer in (6.74) nur Vektoren aus $[0, 1]^k$, so dass deren (komponentenweise definiertes) Maximum $\bigvee_{\nu=1}^r$ auch in $[0, 1]^k$ liegt.

Die komponentenweise Betrachtung der Transformation

$$\begin{aligned} T : [0, 1]^k &\rightarrow [0, 1]^k \\ T(\omega) &= \bigvee_{\nu=1}^r \{ \mu_C(\pi^{\nu}) \wedge [P_{\bar{T}}(\pi^{\nu}) \cdot \omega + P_T(\pi^{\nu}) \cdot \mu_G] \} \end{aligned} \quad (6.77)$$

ergibt

$$\begin{aligned} T_i(\omega) &= \mu_C(\pi^1(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \pi^1(\sigma_i)) \omega_j + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \pi^1(\sigma_i)) \mu_G(\sigma_j) \right] \\ &\vee \dots \\ &\vee \mu_C(\pi^r(\sigma_i) | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \pi^r(\sigma_i)) \omega_j + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \pi^r(\sigma_i)) \mu_G(\sigma_j) \right] \end{aligned} \quad (6.78)$$

für $i = 1, \dots, k$. Weil alle $\pi(\sigma_i) \in \{\alpha_1, \dots, \alpha_m\}$, $i = 1, \dots, k$ und die Maximumbildung idempotent ist, können die Terme der rechten Seite in (6.78) nach α_i , $i = 1, \dots, m$ geordnet zusammengefasst werden, das heißt

36. Als Teilmenge eines Banachraumes ist Ω vollständig.

$$T_i(\omega) = \bigvee_{\nu=1}^m \mu_C(\alpha_\nu | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \alpha_\nu) \omega_j + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \alpha_\nu) \mu_G(\sigma_j) \right] \quad (6.79)$$

für $i = 1, \dots, k$.

Zur Verbesserung der Übersichtlichkeit sei

$$\begin{aligned} c_i(\nu) &:= \mu_C(\alpha_\nu | \sigma_i), \quad i = 1, \dots, k; \\ a_{ij}(\nu) &:= p(\sigma_j | \sigma_i, \alpha_\nu), \quad i, j = 1, \dots, k; \end{aligned} \quad (6.80)$$

$$A(\nu) := (a_{ij}(\nu))_{i,j=1,\dots,k};$$

$$b_i(\nu) := \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \alpha_\nu) \cdot \mu_G(\sigma_j), \quad i = 1, \dots, k$$

und

$$T(\omega) = (T_1(\omega), \dots, T_k(\omega))^T$$

und damit

$$T(\omega) = \begin{pmatrix} T_1(\omega) \\ \dots \\ T_k(\omega) \end{pmatrix} = \bigvee_{\nu=1}^m \left\{ \begin{pmatrix} c_1(\nu) \\ \dots \\ c_k(\nu) \end{pmatrix} \wedge \left[\begin{pmatrix} a_{11}(\nu) & \dots & a_{1k}(\nu) \\ \vdots & \ddots & \vdots \\ a_{k1}(\nu) & \dots & a_{kk}(\nu) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \dots \\ \omega_k \end{pmatrix} + \begin{pmatrix} b_1(\nu) \\ \dots \\ b_k(\nu) \end{pmatrix} \right] \right\}.$$

Die oben beschriebene Menge Ω besitzt die vom *Weissingerschen Fixpunktsatz* (Satz 6.8) vorausgesetzten Eigenschaften. Um ihn anwenden zu können, ist noch die Gültigkeit von (6.58) für die hier betrachtete Transformation T und die Konvergenz der entsprechenden Reihe $\sum \alpha_n$ zu zeigen.

Zunächst wird durch vollständige Induktion nach n bewiesen, dass für alle T_i , $i = 1, \dots, k$ und alle $n = 0, 1, \dots$ folgende Abschätzung gilt:

$$|T_i^n(\omega') - T_i^n(\omega'')| \leq \bigvee_{\nu=1}^m \sum_{j=1}^k a_{ij}^{(n)}(\nu) |\omega_j' - \omega_j''|. \quad (6.81)$$

Für $n = 0$ sei $T^0 := I$ die identische Abbildung $I(\omega) = \omega$ für alle $\omega \in \Omega$. Mit $A^0(\nu) = E$ gilt für alle $\nu = 1, \dots, m$: $a_{ij}^{(0)}(\nu) = \delta_{ij}$. Dann ist

$$\begin{aligned} |T_i^0(\omega') - T_i^0(\omega'')| &= |I_i(\omega') - I_i(\omega'')| = |\omega_i' - \omega_i''| \\ &\leq \sum_{j=1}^k \delta_{ij} |\omega_j' - \omega_j''| = \bigvee_{\nu=1}^m \sum_{j=1}^k a_{ij}^{(0)}(\nu) |\omega_j' - \omega_j''|, \end{aligned}$$

wobei δ_{ij} das Kronecker-Symbol³⁷⁾ ist.

Nun wird angenommen, dass (6.81) für alle $i = 1, \dots, k$ und ein beliebiges, aber festes $n > 0$ gelte. Dann ist

$$\begin{aligned} & |T_i^{n+1}(\omega') - T_i^{n+1}(\omega'')| = |T_i[T^n(\omega')] - T_i[T^n(\omega'')]| \\ & = \left| \bigvee_{\nu=1}^m \left\{ c_i(\nu) \wedge \left[\sum_{j=1}^k a_{ij}(\nu) T_j^n(\omega') + b_i(\nu) \right] \right\} - \bigvee_{\nu=1}^m \left\{ c_i(\nu) \wedge \left[\sum_{j=1}^k a_{ij}(\nu) T_j^n(\omega'') + b_i(\nu) \right] \right\} \right| \end{aligned}$$

und nach Anwendung von Hilfssatz 6.18 und Hilfssatz 6.7 sowie der Dreiecksungleichung:

$$\begin{aligned} |T_i^{n+1}(\omega') - T_i^{n+1}(\omega'')| & \leq \bigvee_{\nu=1}^m \left| \sum_{j=1}^k a_{ij}(\nu) [T_j^n(\omega') - T_j^n(\omega'')] \right| \\ & \leq \bigvee_{\nu=1}^m \sum_{j=1}^k a_{ij}(\nu) |T_j^n(\omega') - T_j^n(\omega'')| \end{aligned}$$

Mit der Induktionsannahme folgt schließlich

$$\begin{aligned} |T_i^{n+1}(\omega') - T_i^{n+1}(\omega'')| & \leq \bigvee_{\nu=1}^m \sum_{j=1}^k a_{ij}(\nu) \left[\sum_{\kappa=1}^k a_{j\kappa}^{(n)}(\nu) |\omega_{\kappa}' - \omega_{\kappa}''| \right] \\ & = \bigvee_{\nu=1}^m \sum_{\kappa=1}^k a_{i\kappa}^{(n+1)}(\nu) |\omega_{\kappa}' - \omega_{\kappa}''| \end{aligned}$$

das heißt, (6.81) gilt auch für $n+1$. Daraus und aus dem Induktionsanfang ($n = 0$) kann geschlossen werden, dass (6.81) für alle $n = 0, 1, \dots$ gilt.

Mit Hilfe von (6.81) lässt sich zeigen, dass für jedes n die Transformation T^n dehnungsbeschränkt im Sinne von (6.58) ist:

$$\|T^n(\omega') - T^n(\omega'')\|_{\infty} = \max_i |T_i^n(\omega') - T_i^n(\omega'')| \leq \max_i \bigvee_{\nu=1}^m \sum_{j=1}^k a_{ij}^{(n)}(\nu) |\omega_j' - \omega_j''|.$$

Wegen der Assoziativität und Kommutativität des Maximum-Operators („max“ bzw. „ \vee “) ist

$$\|T^n(\omega') - T^n(\omega'')\|_{\infty} \leq \bigvee_{\nu=1}^m \max_i \sum_{j=1}^k a_{ij}^{(n)}(\nu) |\omega_j' - \omega_j''|$$

37. siehe Fußnote 5 auf Seite 121

und aufgrund der Verträglichkeit der Matrixnorm $\|\cdot\|_\infty$ und der Vektornorm $\|\cdot\|_\infty$ gilt mit $\tilde{\omega}_j = |\omega_j' - \omega_j''|$:

$$\begin{aligned} \|T^n(\omega') - T^n(\omega'')\|_\infty &\leq \bigvee_{\nu=1}^m \max_{j=1}^k \sum_{i=1}^m a_{ij}^{(n)}(\nu) \tilde{\omega}_j \\ &= \bigvee_{\nu=1}^m \|A^n(\nu) \tilde{\omega}\|_\infty \leq \bigvee_{\nu=1}^m \|A^n(\nu)\|_\infty \|\tilde{\omega}\|_\infty = \bigvee_{\nu=1}^m \|A^n(\nu)\|_\infty \|\omega' - \omega''\|_\infty \end{aligned}$$

woraus schließlich mit

$$\alpha_n := \bigvee_{\nu=1}^m \|A^n(\nu)\|_\infty \quad (6.82)$$

die Dehnungsbeschränktheit nach (6.58) folgt.

Jetzt bleibt zu zeigen, dass die Reihe $\sum_{n=0}^{\infty} \alpha_n$ konvergiert. Nach Voraussetzung ist $\Pi = \Pi_p$ und deshalb gilt $\|P_T^{K(\nu)}(\nu)\| < 1$ mit $1 \leq K(\nu) \leq k$ für alle $\nu = 1, \dots, m$. Daher kann die Reihe in Teilsummen der Länge $K = \max_{\nu} K(\nu)$ zerlegt werden, so dass $\|A^K(\nu)\|_\infty < 1$ mit $1 \leq K \leq k$ für alle $\nu = 1, \dots, m$:

$$\sum_{n=0}^{\infty} \alpha_n = \sum_{n=0}^{\infty} \left[\bigvee_{\nu=1}^m \|A^n(\nu)\|_\infty \right] = \sum_{n=0}^{\infty} \sum_{i=0}^{K-1} \bigvee_{\nu=1}^m \|A^{nK+i}(\nu)\|_\infty. \quad (6.83)$$

Weil A die Untermatrix einer stochastischen Matrix ist, gilt $\|A\|_\infty \leq 1$. Damit folgt, dass die Reihenglieder schwach monoton abnehmen, denn es gilt (vgl. Fußnote 24 auf Seite 267)

$$\bigvee_{\nu=1}^m \|A^{n+1}(\nu)\|_\infty = \bigvee_{\nu=1}^m \|A(\nu) A^n(\nu)\|_\infty \leq \bigvee_{\nu=1}^m (\|A(\nu)\|_\infty \|A^n(\nu)\|_\infty) \leq \bigvee_{\nu=1}^m \|A^n(\nu)\|_\infty$$

Dies erlaubt eine Abschätzung von (6.83) nach oben:

$$\sum_{n=0}^{\infty} \alpha_n \leq \sum_{n=0}^{\infty} K \left[\bigvee_{\nu=1}^m \|A^{nK}(\nu)\|_\infty \right] \leq K \sum_{n=0}^{\infty} \left[\bigvee_{\nu=1}^m (\|A^K(\nu)\|_\infty^n) \right]. \quad (6.84)$$

Weiterhin ist

$$\bigvee_{\nu=1}^m (\|A^K(\nu)\|_\infty^n) = \left(\bigvee_{\nu=1}^m \|A^K(\nu)\|_\infty \right)^n = \left(\bigvee_{\nu=1}^m \|P_T^K(\nu)\|_\infty \right)^n =: q^n$$

mit $q < 1$. Somit folgt in (6.84) die Konvergenz der rechten Seite der Ungleichung und damit

die Konvergenz von $\sum_{n=0}^{\infty} \alpha_n$.

Jetzt sind alle Voraussetzungen des *Weissingerschen Fixpunktsatzes* (Satz 6.8) gezeigt und es folgt, dass T genau einen Fixpunkt $\hat{\omega}$ in Ω besitzt und dieser Fixpunkt der Grenzwert der Iterationsfolge $(T^n \omega^{(0)})$ bei beliebigem $\omega^{(0)} \in \Omega$ ist.

Damit ist Aussage 1 des Satzes gezeigt. Aus (6.72) und (6.73) ist ersichtlich, dass $\mu_D(\pi^{\text{opt}})$ ein Fixpunkt von T ist. Wegen der Eindeutigkeit des Fixpunkts folgt daraus Aussage 2 des Satzes. Aussage 3 geht von der Abschätzung des *Weissingerschen Fixpunktsatzes*, Satz 6.8 aus:

$$d(\tilde{\omega}, T^n \omega^{(0)}) \leq \left(\sum_{\nu=n}^{\infty} \alpha_{\nu} \right) d(T \omega^{(0)}, \omega^{(0)}).$$

Mit (6.84), $\alpha_n := \bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}$ ist zu zeigen, dass $\sum_{\nu=n}^{\infty} \alpha_{\nu} \leq K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}}$ gilt.

Das lässt sich durch aufeinanderfolgende Abschätzungen nach oben und die Anwendung von Hilfssatz 6.19 zeigen. Es ist:

$$\sum_{\nu=n}^{\infty} \alpha_{\nu} = \sum_{\nu=n}^{\infty} [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] = \sum_{\nu=0}^{\infty} [\bigvee_{\mu=1}^m \|A^{n+\nu}(\mu)\|_{\infty}]$$

Nach Anwendung von Hilfssatz 6.19 folgt:

$$\begin{aligned} \sum_{\nu=n}^{\infty} \alpha_{\nu} &\leq \sum_{\nu=0}^{\infty} [\bigvee_{\mu=1}^m (\|A^n(\mu)\|_{\infty} \cdot \|A^{\nu}(\mu)\|_{\infty})] \leq [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] \cdot \sum_{\nu=0}^{\infty} [\bigvee_{\mu=1}^m \|A^{\nu}(\mu)\|_{\infty}] \\ &= [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] \cdot \sum_{\nu=0}^{\infty} \sum_{\lambda=0}^{K-1} [\bigvee_{\mu=1}^m \|A^{\nu K + \lambda}(\mu)\|_{\infty}], \end{aligned}$$

und nach nochmaliger Anwendung von Hilfssatz 6.19:

$$\sum_{\nu=n}^{\infty} \alpha_{\nu} \leq [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] \cdot \sum_{\nu=0}^{\infty} \sum_{\lambda=0}^{K-1} ([\bigvee_{\mu=1}^m \|A^{\nu K}(\mu)\|_{\infty}] \cdot [\bigvee_{\mu=1}^m \|A^{\lambda}(\mu)\|_{\infty}]).$$

Da $\|A^{\lambda}(\mu)\|_{\infty} \leq 1$ folgt $\sum_{\lambda=0}^{K-1} [\bigvee_{\mu=1}^m \|A^{\lambda}(\mu)\|_{\infty}] \leq K$ und damit:

$$\sum_{\nu=n}^{\infty} \alpha_{\nu} \leq [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] \cdot K \sum_{\nu=0}^{\infty} [\bigvee_{\mu=1}^m \|A^{\nu K}(\mu)\|_{\infty}]$$

und nach nochmaliger Anwendung von Hilfssatz 6.19:

$$\sum_{\nu=n}^{\infty} \alpha_{\nu} \leq [\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}] \cdot K \sum_{\nu=0}^{\infty} [\bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}]^{\nu}.$$

Und weil $\|A^K(\mu)\|_{\infty} < 1$ für beliebige μ konvergiert die geometrische Reihe, und es ist:

$$\sum_{\nu=n}^{\infty} \alpha_{\nu} \leq K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}}$$

■

Korollar 6.20.1: Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π .

Wenn die Menge der Politiken Π nur zulässige Politiken enthält, das heißt $\Pi = \Pi_p$, dann gilt für die Transformation T aus (6.74):

Wenn für ein gegebenes $\varepsilon > 0$ gilt:

$$\|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty} < \varepsilon \text{ und } \|T(\omega^{(\nu)}) - \omega^{(\nu)}\|_{\infty} \geq \varepsilon \text{ für } \nu = 0, 1, \dots, n-1,$$

dann kann der Abstand der n -ten Iteration vom Fixpunkt wie folgt abgeschätzt werden:

$$\|T(\omega^{(n)}) - \tilde{\omega}\|_{\infty} < \left[1 + K \frac{\bigvee_{\mu=1}^m \|A(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}} \right] \cdot \varepsilon \leq \left[1 + \frac{K}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}} \right] \cdot \varepsilon,$$

für $\omega^{(n)} := T^n(\omega)$ und $\tilde{\omega}$ ist Fixpunkt von T

sowie $K := \max_{\mu=1, \dots, m} K(\mu)$ mit $1 \leq K(\mu) \leq k$, $k = |\bar{T}|$, so dass

$$\|A^{K(\mu)}(\mu)\|_{\infty} < 1 \text{ für } A(\mu) = P_{\bar{T}}(\pi) \text{ mit } p(\sigma_j | \sigma_i, a_{\mu}) = p(\sigma_j | \sigma_i, \pi(\sigma_i)). \quad \square$$

Beweis: Die Voraussetzungen von Satz 6.20 sind gegeben. Deshalb gilt:

$$\|\tilde{\omega} - T(\omega^{(n)})\|_{\infty} \leq K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}} \cdot \|T(\omega^{(0)}) - \omega^{(0)}\|_{\infty}. \quad (6.85)$$

Für die Norm $\|\cdot\|_{\infty}$ gilt die erweiterte Dreiecksungleichung und deshalb:

$$\|\tilde{\omega} - T(\omega^{(n)})\|_{\infty} = \|[\tilde{\omega} - \omega^{(n)}] - [T(\omega^{(n)}) - \omega^{(n)}]\|_{\infty} \geq \|\tilde{\omega} - \omega^{(n)}\|_{\infty} - \|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty}$$

oder

$$\|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty} + \|\tilde{\omega} - T(\omega^{(n)})\|_{\infty} \geq \|\tilde{\omega} - \omega^{(n)}\|_{\infty}. \quad (6.86)$$

Mit

$$L(n) := K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_{\infty}}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}}$$

ist

$$\|\tilde{\omega} - T(\omega^{(n)})\|_{\infty} \leq L(1) \cdot \|T(\omega^{(n-1)}) - \omega^{(n-1)}\|_{\infty}. \quad (6.87)$$

Mit (6.86) und (6.87) folgt die Abschätzung

$$\|\tilde{\omega} - \omega^{(n)}\|_{\infty} \leq \|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty} + L(1) \cdot \|T(\omega^{(n-1)}) - \omega^{(n-1)}\|_{\infty} < (1 + L(1))\varepsilon$$

■

Die Umsetzung des Verfahrens in ein Programm ist im Abschnitt 6.6.3 beschrieben. Ein Rechenbeispiel wird im Abschnitt 6.6.5 vorgestellt.

Anmerkung zu Satz 6.20 und Korollar 6.20.1

Die Voraussetzung $\Pi = \Pi_p$ ist theoretisch gesehen nicht trivial. Aus praktischer Sicht aber keine wesentliche Einschränkung. Näheres dazu in Abschnitt 6.5.3.1.

6.4.4 Verallgemeinerung für andere t-Normen

Alle Aussagen der vorigen Abschnitte gehen davon aus, dass unscharfe Nebenbedingungen und unscharfe Ziele durch die Minimumsbildung als Aggregationsoperator verbunden werden. Dies entspricht der Behandlung der unscharfen dynamischen Optimierung in dem grundlegenden Artikel von Bellman und Zadeh [11], wie auch die Wahl des Minimumsoperators unter bestimmten Annahmen zwingend ist [12]. In den Anmerkungen auf Seite 245 wird auf einen allgemeineren Ansatz von Zimmermann hingewiesen, der für reale Anwendungen flexibler ist (vgl. Def. 6.7). Er ist nicht auf t-Normen beschränkt und erlaubt die Benutzung unterschiedlicher Aggregationsoperatoren. In diesem Unterabschnitt wird untersucht, ob die Hauptaussagen der vorigen Abschnitte für beliebige t-Normen gültig bleiben bzw. welche Eigenschaften eine t-Norm besitzen muss, damit dies der Fall ist. Im einzelnen ist folgendes zu klären:

1. Inwiefern ist die algebraische Struktur der Politik-Faktormenge Π/\sim von der t-Norm abhängig?

Die Frage betrifft die Halbordnung $(\Pi/\sim, \geq)$ (vgl. Satz 6.14) und die Aussagen der Sätze 6.16 und 6.17, dass Π/\sim ein endlicher, vollständiger Verband ist und es eine eindeutige optimale Politik-Äquivalenzklasse gibt.

2. Inwiefern sind die Hauptaussagen

- über die iterative Berechnung des unscharfen Entscheidungsvektors $\mu_D(\pi)$ einer gegebenen Politik $\pi \in \Pi$ (Satz 6.9),
 - über die iterative Berechnung einer optimalen Politik $\pi^{\text{opt}} \in \Pi$ (Satz 6.20) und ihre Fehlerabschätzung (Korollar 6.20.1)
- für t-Normen im Allgemein gültig.

Diese Fragen werden im Folgenden unter der Annahme geklärt, dass alle unscharfen Nebenbedingungen und unscharfen Ziele durch die gleiche t-Norm verknüpft sind, es gibt also keine Differenzierung wie in Def. 6.7.

6.4.4.1 Erweiterung der Grundlagen

Von zentraler Bedeutung ist Hilfssatz 6.7. Seine Aussage wird in den Beweisen der beiden Hauptsätze 6.9 und 6.20 benötigt. Ohne sie kann der Weissingersche Fixpunktsatz (Satz 6.8) nicht angewandt werden. Hilfssatz 6.7 besagt:

Für alle $u, v, w \in \mathbb{R}$ und die *Minimum-t-Norm*

$$t_{\min}(u, v) = \min\{u, v\} \quad (6.88)$$

$$\text{gilt } |t_{\min}(u, v) - t_{\min}(u, w)| \leq |v - w|.$$

Andere t-Normen besitzen im Allgemeinen nicht diese Eigenschaft. Ein Beispiel ist das so genannte *drastische Produkt* („drastic product“),

$$t_{\text{dp}}(u, v) = \begin{cases} u & \text{falls } v = 1 \\ v & \text{falls } u = 1 \\ 0 & \text{sonst} \end{cases} \quad (6.89)$$

Mit $u = 0,8$, $v = 0,9$, $w = 1$ ist nämlich einerseits

$$|t_{\text{dp}}(u, v) - t_{\text{dp}}(u, w)| = |0 - 0,8| = 0,8,$$

aber andererseits

$$|v - w| = |0,9 - 0,8| = 0,1.$$

Andererseits besitzt die t-Norm *algebraisches Produkt* („algebraic product“),

$$t_{\text{ap}}(u, v) = u \cdot v \quad (6.90)$$

diese Eigenschaft, denn offensichtlich ist für alle $u, v, w \in [0, 1]$

$$|u \cdot v - u \cdot w| = |u \cdot (v - w)| = |u| \cdot |v - w| \leq |v - w|.$$

Aus den Beweisen der Hauptsätze 6.9 und 6.20 ist ersichtlich, dass die t-Norm eine Lipschitzbedingung bezüglich eines Arguments erfüllen muss, das heißt die t-Norm t besitzt für beliebige $u, v, w \in [0, 1]$ die Eigenschaft:

$$\exists L \geq 0 : |t(u, v) - t(u, w)| \leq L|v - w|, \quad (6.91)$$

wobei insbesondere L nicht von u , v oder w abhängt. Damit können die Hauptsätze 6.9 und 6.20 verallgemeinert werden. Die Formulierung folgt in Abschnitt 6.4.4.2. Der Beweis von Korollar 6.20.1 beruht vollständig auf der Aussage von Satz 6.20 und bleibt deshalb erhalten. Damit ist die einleitende **Frage 2** beantwortet.

Die Klärung von **Frage 1** beruht alleine auf Hilfssatz 6.15:

- Die Halbordnungseigenschaft von $(\Pi/\sim, \geq)$ ist vom Operator T (und seiner t -Norm) unabhängig (vgl. Hilfssätze 6.10 und 6.12 und Satz 6.14).
- Die von Satz 6.16 behauptete Verbandseigenschaft von $(\Pi/\sim, \geq)$ hängt von der Wahl der t -Norm im Operator T nur in der Aussage über die Existenz von Infimum und Supremum für jeweils zwei Politik-Äquivalenzklassen aus Π/\sim ab. Sie kann mittels Hilfssatz 6.15 bewiesen werden. Die Aussage von Satz 6.17 wiederum ist genau dann wahr, wenn Satz 6.16 gilt.

Zum Beweis von Hilfssatz 6.15 für eine andere t -Norm als die Minimumsbildung muss Satz 6.9 für die gewählte t -Norm gültig sein, also die Lipschitzbedingung (6.91) gelten bzw. die Verallgemeinerung von Satz 6.9 (siehe nächster Abschnitt). Ansonsten wird im Beweis nur noch die Monotonie der t -Norm benötigt, die aber per Definition (Def. 2.7) eine Eigenschaft aller t -Normen ist. Damit können auch Hilfssatz 6.15 und die Sätze 6.16 und 6.17 verallgemeinert werden.

In den Fallbeispielen in Abschnitt 6.6 werden außer der Minimum- t -Norm und den oben untersuchten t -Normen: algebraisches Produkt und drastisches Produkt auch die t -Normen

- *beschränkte Differenz* („bounded difference“)

$$t_{bd}(u, v) = \max\{0, u + v - 1\}, \quad (6.92)$$

- *Hamacher-Produkt* („Hamacher product“)

$$t_{hp}(u, v) = \begin{cases} \frac{u \cdot v}{u + v - u \cdot v} & \text{falls } u, v \neq 0 \\ 0 & \text{sonst} \end{cases}, \quad (6.93)$$

- *Einstein-Produkt* („Einstein product“)

$$t_{ep}(u, v) = \frac{u \cdot v}{1 + (1 - u) \cdot (1 - v)} \quad (6.94)$$

untersucht. Ob diese t -Normen die Lipschitzbedingung (6.91) erfüllen lässt sich einfacher mit Hilfe des folgenden Satzes beantworten.

Satz 6.21: Besitzt die Funktion f auf dem beliebigen Intervall I eine *beschränkte* Ableitung, so ist sie *Lipschitz-stetig* auf I mit der Lipschitz-Konstanten

$$L := \sup_{x \in I} |f'(x)|.$$

□

Beweis: siehe [71]

Weil t -Normen nach Def. 2.7 kommutativ sind, ist es somit ausreichend für ein beliebiges aber festes $u \in [0, 1]$, die Funktion $f(v) := t(u, v)$ zu betrachten und ihre Ableitung $f'(v)$ für $v \in [0, 1]$ zu untersuchen.

Beschränkte Differenz

Für die beschränkte Differenz (6.92) und $u \in (0, 1)$ besitzt $f(v) = \max\{0, u + v - 1\}$ offensichtlich in $v = 1 - u$ keine Ableitung: Für $v \in [0, 1 - u]$ ist $f(v) \equiv 0$, also $f'(v) = 0$. Für $v \in [1 - u, 1]$ ist $f(v) = u + v - 1$, also $f'(v) = 1$. Da f beschränkt ist, existieren an den Intervallgrenzen 0 und $1 - u$ jeweils einseitige Ableitungen. Damit ist der Satz 6.21 für $v, w \in [0, 1 - u]$ und für $v, w \in [1 - u, 1]$ anwendbar. Wie verhält es sich, wenn

$$v \in [0, 1 - u) \text{ und } w \in [1 - u, 1] ? \quad (6.95)$$

Sei L_1 die Lipschitzkonstante von f in $[0, 1 - u)$ und L_2 die Lipschitzkonstante von f in $[1 - u, 1]$ und o. B. d. A. $L := \max\{L_1, L_2\}$. Dann gilt für (6.95)

$$\begin{aligned} |f(v) - f(w)| &= f(w) - f(v) = [f(w) - f(1 - u)] + [f(1 - u) - f(v)] \\ &\leq L_2(w - 1 + u) + L_1(1 - u - v) \leq L(w - 1 + u + 1 - u - v) = L(w - v) \\ &= L|v - w| \end{aligned}$$

und somit ist f in $[0, 1]$ Lipschitz-stetig.

Hamacher-Produkt

Für $u = 0$ ist die Funktion $f(v) := \mathbf{t}_{\text{hp}}(u, v)$ mit $\mathbf{t}_{\text{hp}}(u, v)$ aus (6.93) identisch null, also Lipschitzstetig. Für $u \neq 0$ ist die Funktion $f(v)$ in $(0, 1)$ differenzierbar mit

$$f'(v) = \frac{u^2}{(u + v - u \cdot v)^2}.$$

An den Intervallgrenzen sind die einseitigen Ableitungen $\lim_{v \rightarrow 0+} f'(v) = 1$ und $\lim_{v \rightarrow 1-} f'(v) = u^2$. Die Ableitung von f ist somit beschränkt auf ganz $[0, 1]$ und somit folgt mit Satz 6.21 die Lipschitz-Stetigkeit von f .

Einstein-Produkt

Für das Einstein-Produkt ist

$$f(v) := \mathbf{t}_{\text{ep}}(u, v) = \frac{u \cdot v}{1 + (1 - u) \cdot (1 - v)}.$$

Die Ableitung in $(0, 1)$ ist

$$f'(v) = \frac{u(2 - u)}{[1 + (1 - u) \cdot (1 - v)]^2}.$$

An den Intervallrändern ist $\lim_{v \rightarrow 0+} f(v) = \frac{u}{2-u}$ und $\lim_{v \rightarrow 1-} f(v) = u(2-u)$. Da alle Ableitungen beschränkt sind, ist Satz 6.21 anwendbar und f Lipschitz-stetig.

Zusammenfassend lässt sich feststellen, dass bis auf das drastische Produkt alle t -Normen die Lipschitz-Bedingung (6.91) erfüllen.

6.4.4.2 Verallgemeinerung der Hauptsätze

Die Beweise der folgenden Sätze entsprechen den Beweisen der korrespondierenden Sätze in den Abschnitten 6.4.2 und 6.4.3, unter Berücksichtigung der Bemerkungen im Abschnitt 6.4.4.1. Zur einfachen Formulierung der Sätze werden zunächst die beiden Transformationen T_π^t und T^t definiert.

Definition 6.17: Transformationen T_π^t und T^t .

Für einen mehrstufigen stochastischen Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 und für die t -Norm t sind die folgenden Transformationen definiert:

$$\begin{aligned} T_\pi^t &: [0, 1]^k \rightarrow [0, 1]^k \\ T_\pi^t(\omega) &:= \mu_C(\pi) \, t[P_{\bar{T}}(\pi) \cdot \omega + P_T(\pi) \cdot \mu_G] \end{aligned} \quad (6.96)$$

für alle $\pi \in \Pi$;

$$\begin{aligned} T^t &: [0, 1]^k \rightarrow [0, 1]^k \\ T^t(\omega) &= \bigvee_{\pi \in \Pi} \{ \mu_C(\pi^\nu) \, t[P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G] \}. \end{aligned} \quad (6.97)$$

□

Satz 6.22: (Verallgemeinerung von Satz 6.9): Gegeben seien ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10, die Menge der Politiken Π und die Transformation T_π^t nach Def. 6.17. Wenn die t -Norm t die Lipschitzbedingung (6.91) erfüllt und die Politik $\pi \in \Pi$ zulässig ist, dann besitzt T_π^t einen eindeutigen Fixpunkt $\tilde{\omega} = \mu_D(\pi)$, und die Iterationsfolge $(\omega^{(n)})$ mit $\omega^{(n+1)} = T_\pi^t(\omega^{(n)})$ konvergiert für beliebige Anfangswerte $\omega^{(0)}$ gegen den Fixpunkt. Dabei gilt die Fehlerabschätzung

$$\|\tilde{\omega} - T_{\pi}^t(\omega^{(n)})\|_{\infty} \leq K \frac{\|P_{\tilde{T}}^n(\pi)\|_{\infty}}{1 - \|P_{\tilde{T}}^K(\pi)\|_{\infty}} \cdot \|T_{\pi}^t(\omega^{(0)}) - \omega^{(0)}\|_{\infty},$$

wobei $1 \leq K \leq k$ so dass $\|P_{\tilde{T}}^K(\pi)\|_{\infty} < 1$. \square

Hilfssatz 6.23: (Verallgemeinerung von Hilfssatz 6.15): Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10, mit der Menge der Politiken Π und ihrer Teilmenge der zulässigen Politiken Π_p sowie der Transformation T_{π}^t nach Def. 6.17. Wenn die t-Norm t die Lipschitzbedingung (6.91) erfüllt, dann gilt:

1. Für jede Politik $\tilde{\pi} \in \Pi_p$ sind durch

$$T_{\pi_g}^t(\mu_D(\tilde{\pi})) := \max_{\pi \in \Pi} T_{\pi}^t(\mu_D(\tilde{\pi})) \quad (6.98)$$

und

$$T_{\pi_k}^t(\mu_D(\tilde{\pi})) := \min_{\pi \in \Pi} T_{\pi}^t(\mu_D(\tilde{\pi})) \quad (6.99)$$

die Politiken $\pi_g, \pi_k \in \Pi$ gegeben.

2. Gilt $\pi_g, \pi_k \in \Pi_p$, dann ist $\pi_g \geq \tilde{\pi}$ und $\pi_k \leq \tilde{\pi}$. \square

Satz 6.24: (Verallgemeinerung von Satz 6.16) Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Halbordnung $(\Pi/\sim, \geq)$ aus Satz 6.14. Die die Halbordnung $(\Pi/\sim, \geq)$ induzierenden Entscheidungsvektoren $\mu_D(\pi)$ seien gemäß Satz 6.22 mit der Transformation T_{π}^t nach Def. 6.17 berechnet.

Wenn die Menge der Politiken Π nur zulässige Politiken enthält, das heißt $\Pi = \Pi_p$ und wenn die t-Norm t die Lipschitz-Bedingung (6.91) erfüllt, dann ist die Politik-Faktormenge Π/\sim ein *endlicher vollständiger Verband*. \square

Satz 6.25: (Verallgemeinerung von Satz 6.17) Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10 mit der Menge der Politiken Π sowie der Halbordnung $(\Pi/\sim, \geq)$ aus Satz 6.14. Die die Halbordnung

$(\Pi/\sim, \geq)$ induzierenden Entscheidungsvektoren $\mu_D(\pi)$ seien gemäß Satz 6.22 mit der Transformation T_π^t nach Def. 6.17 berechnet.

Wenn die Menge der Politiken Π nur zulässige Politiken enthält, das heißt $\Pi = \Pi_p$ und wenn die t-Norm t die Lipschitz-Bedingung (6.91) erfüllt, dann gibt es eine eindeutige Politik-Äquivalenzklasse $[\tilde{\pi}]$, die nur optimale Politiken enthält: $\tilde{\pi} \sim \pi^{\text{opt}}$. Sie wird *optimale Politik-Äquivalenzklasse* genannt. \square

Satz 6.26: (Verallgemeinerung von Satz 6.20): Gegeben sei ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit, (S, X, T, U, P, C, G) nach Def. 6.10, der nur zulässige Politiken besitze, das heißt $\Pi = \Pi_p$, sowie die Transformation T^t nach Def. 6.17. Wenn die t-Norm t die Lipschitzbedingung (6.91) erfüllt, dann gilt:

1. Das Fixpunktproblem $T^t(\omega) = \omega$ besitzt genau eine Lösung, die durch die Iteration $\omega^{(k+1)} = T^t(\omega^{(k)})$ für beliebiges $\omega^{(0)} \in [0, 1]^k$ berechnet werden kann.
2. Die Lösung des Fixpunktproblems ist der eindeutige unscharfe Entscheidungsvektor einer optimalen Politik $\mu_D(\pi^{\text{opt}})$.
3. Es gilt die Fehlerabschätzung

$$\|\tilde{\omega} - T^t(\omega^{(n)})\|_\infty \leq K \frac{\bigvee_{\mu=1}^m \|A^n(\mu)\|_\infty}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_\infty} \cdot \|T^t(\omega^{(0)}) - \omega^{(0)}\|_\infty,$$

wobei $K := \max_{\mu=1, \dots, m} K(\mu)$ mit $1 \leq K(\mu) \leq k$, $k = |\bar{T}|$, so dass $\|A^{K(\mu)}(\mu)\|_\infty < 1$ für $A(\mu) = P_{\bar{T}}(\pi)$ mit $p(\sigma_j | \sigma_i, a_\mu) = p(\sigma_j | \sigma_i, \pi(\sigma_i))$. \square

6.5 Praktische Berechnung einer optimalen Politik

Im vorigen Abschnitt ist die iterative Berechnung

- des unscharfen Entscheidungsvektors für eine gegebene zulässige Politik und
- des unscharfen Entscheidungsvektors einer optimalen Politik

hergeleitet worden. Darauf aufbauend beschreiben die folgenden Abschnitte, wie eine optimale Politik praktisch berechnet werden kann. Im Abschnitt 6.5.1 wird die vollständige Aufzählung aller Politiken beschrieben und im Abschnitt 6.5.2 wird gezeigt, wie die optimale Politik, aufbauend auf der Iteration aus Abschnitt 6.4.3 berechnet wird. Zu den Verfahren werden Algorithmen in Pseudo-Code angegeben. Alle beinhalten das Unterprogramm *Initialisierung* (Alg. 6.1).

```

Definiere Initialisierung(...) {
  Für jeden nicht beendeten Zustand  $x_t = \sigma_1, \dots, \sigma_k$  führe aus {
    Für jede Entscheidung  $u_t = \alpha_1, \dots, \alpha_m$  führe aus
      Eingabe  $\mu_C(u_t | x_t)$ ;
      Für jeden Zustand  $x_{t+1} = \sigma_1, \dots, \sigma_n$  führe aus
        Eingabe  $p(x_{t+1} | x_t, u_t)$ ;
      }
    Für jeden beendeten Zustand  $x_{t+1} = \sigma_{k+1}, \dots, \sigma_n$  führe aus
      Eingabe  $\mu_G(x_{t+1})$ ;
  }
}

```

Algorithmus 6.1: Unterprogramm *Initialisierung*.

6.5.1 Optimierung durch vollständige Aufzählung

Die Menge der Politiken ist endlich. Wie an verschiedenen Stellen bereits erwähnt, gibt es für eine Zustandsmenge X mit k nicht beendeten Zuständen und eine Entscheidungsmenge U mit m Entscheidungen $r = m^k$ unterschiedliche Politiken. Also ist

$$\Pi = \{\pi^1, \pi^2, \dots, \pi^r\}.$$

Somit kann die optimale Politik durch Probieren aller Politiken ermittelt werden. Wie Alg. 6.2

```

Führe aus
  Initialisierung(...);
Setze  $\pi^{\text{opt}} := \pi^1$ ;
Für  $i = 2, \dots, r$  führe aus
  Wenn  $\pi^i > \pi^{\text{opt}}$  dann  $\pi^{\text{opt}} := \pi^i$ ;
Ausgabe:  $\pi^{\text{opt}}$  ist die optimale Politik;

```

Algorithmus 6.2: „Naive“ vollständige Aufzählung aller Politiken.

zeigt, müssen dafür jeweils zwei Politiken verglichen werden. Weil die Politikmenge mit der „besser als“-Relation aus Def. 6.13 nur eine Quasi-Ordnung bildet, braucht die „Wenn“-Abfrage im Algorithmus weder wahr noch falsch zu sein. Daher wird er als „naive“ vollständige Aufzählung bezeichnet. Sie kann nicht praktisch realisiert werden. Dazu kommt noch, dass Politiken, also Abbildungen aus der Zustandsmenge in die Entscheidungsmenge, $\pi : X \rightarrow U$ nicht direkt verglichen werden können, sondern gemäß Def. 6.13 nur ihre unscharfen Entscheidungsvektoren $\mu_D(\pi)$. Diese existieren aber nur für zulässige Politiken, so dass $\Pi = \Pi_p$ vorausgesetzt werden muss. Mit dieser Annahme ist es gemäß Hilfssatz 6.15 möglich, für jeweils zwei Politiken π' und π'' eine bessere Politik π_g zu finden, also $\pi_g \geq \pi', \pi''$. Damit ist die vollständige Aufzählung aller Politiken realisierbar (siehe Alg. 6.3).

```

Führe aus
  Initialisierung(...);
Setze  $\pi^{\text{opt}} := \pi^1$ ,  $\text{opt} := \text{'falsch'}$ ,  $i := 1$ ;
Berechne  $\mu_D(\pi^{\text{opt}})$ ; // Satz 6.9
Für  $i = 2, \dots, r$  führe aus {
  Berechne  $\mu_D(\pi^i)$ ; // Satz 6.9
  Wenn  $\mu_D(\pi^i) \geq \mu_D(\pi^{\text{opt}})$  dann // Def. 6.13
    Setze  $\pi^{\text{opt}} := \pi^i$ ,  $\mu_D(\pi^{\text{opt}}) := \mu_D(\pi^i)$ ;
  Sonst führe aus {
    Berechne  $\mu_D^{\text{max}}$  aus  $\mu_D(\pi^{\text{opt}})$  und  $\mu_D(\pi^i)$ ; // Glg. 6.70
    Berechne  $\pi_g$  mit  $\pi_g \geq \pi^{\text{opt}}, \pi^i$ ; // Hilfssatz 6.15
    Berechne  $\mu_D(\pi_g)$ ; // Satz 6.9
    Setze  $\pi^{\text{opt}} := \pi_g$ ,  $\mu_D(\pi^{\text{opt}}) := \mu_D(\pi_g)$ ;
  }
}
Ausgabe:  $\pi^{\text{opt}}$  ist eine optimale Politik;

```

Algorithmus 6.3: Optimierung durch vollständige Aufzählung aller Politiken.

6.5.2 Optimierung durch Iteration

Gänzlich ohne Anwendung von Hilfssatz 6.15 kommt die Optimierung aus, wenn der unscharfe Entscheidungsvektor durch Iteration gemäß Satz 6.20 berechnet wird. Die Transformation (6.74) wird durch einfache Überlegung in die Form

$$T_i(\omega) = \bigvee_{\nu=1}^m \mu_C(\alpha_\nu | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \alpha_\nu) \omega_j + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \alpha_\nu) \mu_G(\sigma_j) \right] \quad (6.100)$$

für $i = 1, \dots, k$ überführt (vgl. Beweis von Satz 6.20). Ausgehend von einem beliebig gewählten Vektor $\omega^{(0)} \in [0, 1]^k$ besteht ein Iterationsschritt in der Anwendung des Operators T . Für jede seiner Komponenten, $i = 1, \dots, k$ ist aus der Entscheidungsmenge U eine Entscheidung $\alpha_i^{(n+1)}$ zu wählen, so dass gilt:

$$\begin{aligned} & \mu_C(\alpha_i^{(n+1)} | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \alpha_i^{(n+1)}) \omega_j^{(n)} + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \alpha_i^{(n+1)}) \mu_G(\sigma_j) \right] \\ &= \bigvee_{\nu=1}^m \mu_C(\alpha_\nu | \sigma_i) \wedge \left[\sum_{\sigma_j \in \bar{T}} p(\sigma_j | \sigma_i, \alpha_\nu) \omega_j^{(n)} + \sum_{\sigma_j \in T} p(\sigma_j | \sigma_i, \alpha_\nu) \mu_G(\sigma_j) \right] \end{aligned} \quad (6.101)$$

für $n = 0, 1, \dots$. Gemäß Satz 6.20 konvergiert die Iteration. Sei

$$\|T(\omega^{(\tilde{n})}) - \omega^{(\tilde{n})}\|_{\infty} < \varepsilon$$

für ein vorgegebenes kleines $\varepsilon > 0$ das Abbruchkriterium der Iteration. Dann sind die Entscheidungen $\alpha_i^{(\tilde{n}+1)}$ die Bilder einer optimalen Politik π^{opt} , das heißt

$$\pi^{\text{opt}}(\sigma_i) = \alpha_i^{(\tilde{n}+1)}. \quad (6.102)$$

Der Algorithmus ist in Alg. 6.4 dargestellt.

```

Führe aus
  Initialisierung(...);
Eingabe  $\omega^{(0)} \in [0, 1]^k$ ,  $\varepsilon > 0$ ; // Iterationsbeginn, Abbruchkriterium
Setze opt := 'falsch', n = 0;
Solange opt = 'falsch' führe aus {
  Für i = 1, ..., k führe aus
    Berechne  $T_i(\omega^{(n)})$ ; // Glg. 6.100
    Speichere  $\alpha_i^{(n+1)}$ ;
  Wenn  $\|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty} < \varepsilon$  dann {
    opt := 'wahr';
    Setze  $\pi^{\text{opt}}$  so dass  $\pi^{\text{opt}}(\sigma_1, \dots, \sigma_k) := (\alpha_1^{(n+1)}, \dots, \alpha_k^{(n+1)})$ ;
    // Glg. 6.101, Glg. 6.102
  }
  Sonst führe aus {
     $\omega^{(n)} := T(\omega^{(n-1)})$ ;
    Setze n = n + 1;
  }
}
Ausgabe:  $\pi^{\text{opt}}$  ist eine optimale Politik;

```

Algorithmus 6.4: Optimierung durch Iteration.

Das Abbruchkriterium $\|T(\omega^{(n)}) - \omega^{(n)}\|_{\infty} < \varepsilon$ für vorgegebenes $\varepsilon > 0$ sagt nichts über den Abstand der Iterierten $T(\omega^{(n)})$ vom Fixpunkt $\tilde{\omega}$ aus, der letztendlich die Güte der Approximation (bei Abbruch) beschreibt. Mit Hilfe von Korollar 6.20.1 kann diese Lücke geschlossen werden, denn aus dem Abbruchkriterium von Alg. 6.4 folgt

$$\|T(\omega^{(n)}) - \tilde{\omega}\|_{\infty} < \left[1 + \frac{K}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_{\infty}} \right] \cdot \varepsilon.$$

Wird für gegebenes $\varepsilon > 0$ das Abbruchkriterium modifiziert: $\|T(\omega^{(n-1)}) - \omega^{(n-1)}\|_\infty < \tilde{\varepsilon}$, mit $\tilde{\varepsilon} := \frac{\varepsilon}{L}$ und $L := 1 + \frac{K}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_\infty}$, dann gilt

$$\|T(\omega^{(n)}) - \tilde{\omega}\|_\infty < \left[1 + \frac{K}{1 - \bigvee_{\mu=1}^m \|A^K(\mu)\|_\infty} \right] \cdot \tilde{\varepsilon} = \varepsilon.$$

Die Berechnung von K ist über einfache Matrizenmultiplikation möglich. Das Unterprogramm zur Berechnung von L zeigt Alg. 6.5 und in Alg. 6.6 wird die Optimierung durch Itera-

```

Definiere Epsilonkonstante( $L$ ) {
  Für jede Politik  $\pi = \pi^1, \dots, \pi^r$  führe aus {
    Setze  $i := 1$ ;
    Berechne  $\|P_T^i(\pi)\|_\infty$ ;
    Solange  $\|P_T^i(\pi)\|_\infty \geq 1$  führe aus {
      Setze  $i := i + 1$ ;
      Berechne  $\|P_T^i(\pi)\|_\infty$ ;
    }
    Setze  $K(\pi) := i$ ;
  }
  Berechne  $K := \max_{\pi = \pi^1, \dots, \pi^r} K(\pi)$ ;
  Für jede Politik  $\pi = \pi^1, \dots, \pi^r$  führe aus
    Berechne  $\|P_T^K(\pi)\|_\infty$ ;
  Berechne  $L' := \max_{\pi = \pi^1, \dots, \pi^r} \|P_T^K(\pi)\|_\infty$ ;
  Berechne  $L := 1 + \frac{K}{1 - L'}$ ;
  Ausgabe:  $L$  ist die Epsilonkonstante;
}

```

Algorithmus 6.5: Unterprogramm *Epsilonkonstante*.

tion mit verbesserter Abbruchbedingung gezeigt.

6.5.3 Zur Erreichbarkeit von Endzuständen und einigen anderen praktischen Fragen

6.5.3.1 Erreichbarkeit von Endzuständen

Im Abschnitt 6.4.2 wird gezeigt, unter welchen Bedingungen ein mehrstufiger stochastischer Entscheidungsprozess in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit einen Endzustand erreichen kann. Unter der Annahme, dass mit jeder Politik ein Endzustand erreichbar ist, sagt Satz 6.20 aus, wie eine optimale Politik berechnet werden kann. In diesem

```

Führe aus {
  Initialisierung(...);
  Epsilonkonstante(L);
}
Eingabe  $\omega^{(0)} \in [0, 1]^k$ ,  $\varepsilon > 0$ ;      // Iterationsbeginn, Abbruchkriterium
Setze  $\text{opt} := \text{'falsch'}$ ,  $n = 0$ ;
Solange  $\text{opt} = \text{'falsch'}$  führe aus {
  Für  $i = 1, \dots, k$  führe aus
    Berechne  $T_i(\omega^{(n)})$ ;                      // Glg. 6.100
    Speichere  $\alpha_i^{(n+1)}$ ;
  Wenn  $\|T(\omega^{(n)}) - \omega^{(n)}\|_\infty < \frac{\varepsilon}{L}$  dann {      // Verbesserung
     $\text{opt} := \text{'wahr'}$ ;
    Setze  $\pi^{\text{opt}}$  so dass  $\pi^{\text{opt}}(\sigma_1, \dots, \sigma_k) := (\alpha_1^{(n+1)}, \dots, \alpha_k^{(n+1)})$ ;
    // Glg. 6.101, Glg. 6.102
  }
  Sonst führe aus {
     $\omega^{(n)} := T(\omega^{(n-1)})$ ;
    Setze  $n = n + 1$ ;
  }
}
Ausgabe:  $\pi^{\text{opt}}$  ist die optimale Politik;

```

Algorithmus 6.6: Optimierung durch Iteration (Alg. 6.4) mit verbesserter Abbruchbedingung.

Zusammenhang stellt sich die Frage, was die Erreichbarkeit eines Endzustandes praktisch bei der Modellierung der Neukundengewinnung bedeutet.

Die Unerreichbarkeit aller Endzustände setzt voraus, dass es unter den nicht-beendeten Zuständen solche gibt, die nicht mehr verlassen werden können (das heißt der Zustand ist *absorbierend*) oder dass Kreise in einer Kette von Zustandsübergängen zwischen nicht-beendeten Zuständen möglich sind, so dass ein Zustand nach Durchlaufen solcher Kreise mit Wahrscheinlichkeit eins wieder erreicht wird (das heißt der Zustand ist *rekurrent*). Das bedeutet, für eine gegebenen Politik π gibt es einen Zustand $\sigma_i \in \bar{T}$, so dass

$$p_{ii}(\pi) = 1, \quad (6.103)$$

(das heißt σ_i ist absorbierend) oder

$$\sum_{\kappa=1}^{\infty} p_{ii}^{(\kappa)}(\pi) = \infty, \quad (6.104)$$

(das heißt σ_i ist rekurrent), wobei $p_{ii}^{(\kappa)}(\pi)$ ein Element der κ -Schritt-Übergangsmatrix ist³⁸). In Korollar 6.6.1 wurde gezeigt, dass für eine zulässige Politik π die Gleichungen (6.103) und (6.104) nicht gelten. Unter welchen Umständen aber in der Praxis beide Fälle auftreten können, ist Gegenstand der nachfolgenden Betrachtung.

Unter den Kundenmerkmalen in den Beispielen in Abschnitt 5.3 können zwei Typen unterschieden werden:

1. Kundenmerkmale, die ihren Wert im Verlauf einer Kundenakquisition nur einmal ändern und ihn dann beibehalten. Dazu gehören Merkmale wie Einkommen, Alter, Bildungsstand. Häufig sind sie zum Beginn der Neukundengewinnung unbekannt und ändern ihren Wert, wenn sie der Akquisiteur in Erfahrung gebracht hat. Danach bleiben sie (über die Dauer des Akquisitionsprozesses) konstant. Solche Merkmale könnten die Ursache für absorbierende Zustände (Kundendatensätze) sein.
2. Kundenmerkmale, die ihren Wert relativ häufig, z. B. bei jedem Zustandsübergang ändern können. Dazu gehört das Interesse des Kunden am Produkt, das zunächst unbekannt sein kann, nach einer Reaktion auf einen Massenbrief als unentschlossen eingestuft wird, nach Anforderung grundlegender Informationen über das Produkt hoch ist und nach dem Erhalt von Detailinformationen wieder den Wert „unentschlossen“ annimmt. Solche Merkmale könnten die Ursache für rekurrente Zustände (Kundendatensätze) sein.

Der erste Fall führt nur dann zu absorbierenden Zuständen, wenn sich *alle* Kundenmerkmale ab einem bestimmten Zustand nicht mehr ändern, was eher unwahrscheinlich ist, da z. B. einige Merkmalswerte auf der subjektiven Einschätzung des Verkäufers/Vertreters o. Ä. beruhen und so nach jedem Zustandsübergang geändert werden können. Der zweite Fall kann zwar zu rekurrenten Zuständen führen, (6.104) muss aber nicht zwangsläufig gelten, so dass ein Endzustand erreichbar sein kann.

Absorbierende oder rekurrente Zustände können auch durch die Einführung der Kundenmerkmale „Neuigkeit“ und „Häufigkeit“ (vgl. Anmerkungen auf Seite 234 im Kapitel 5) nicht ausgeschlossen werden. Bei einem Kunden, der auf alle Akquiseaktionen reagiert bleibt der Wert für das Merkmal „Neuigkeit“ gleich null und wenn „Häufigkeit“ ein Verhältnisswert, wie in (5.4) auf Seite 235 definiert ist, dann bleibt der Wert für das Merkmal „Häufigkeit“ gleich eins.

Anders ist die Situation, wenn „Häufigkeit“ über absolute Werte definiert ist, das heißt die Häufigkeit der Kundenreaktion ist gleich der Anzahl der Kundenreaktionen. Dann können keine zwei gleichen Zustände hintereinander auftreten. In Abb. 6.2 ist die Projektion des Zustandsraumes in die Ebene, die von den Merkmalen „Neuigkeit“,

$$\xi_{p+1} \in S_{p+1} = \{0, 1, \dots, N_{p+1}\}$$

und „Häufigkeit“

$$\xi_{p+2} \in S_{p+2} = \{0, 1, \dots, N_{p+2}\}$$

38. Somit können absorbierende Zustände als Sonderfälle von rekurrenten Zuständen betrachtet werden; siehe auch Anmerkungen auf Seite 263.

aufgespannt wird, dargestellt [62]. Dabei nehmen die Merkmalsvariablen ξ_{p+1} und ξ_{p+2} Werte aus den Merkmalsmengen S_{p+1} bzw. S_{p+2} , wobei p die Anzahl der Merkmale vor dieser Erweiterung ist. Die Werte N_{p+1} und N_{p+2} sind Erfahrungswerte des Akquisiteurs und ausreichend groß gewählt. Weiterhin wird angenommen, dass S_{p+1} und S_{p+2} so partitioniert sind, dass jede Partition aus genau einem Wert besteht. Mit dem Anfangswert

$$x_{p+1}(0) = x_{p+2}(0) = 0$$

gilt für alle Folgezustände:

$$x_{p+1}(t+1) = \begin{cases} 0 & \text{falls Kunde reagiert} \\ x_{p+1}(t) + 1 & \text{falls Kunde reagiert nicht} \end{cases},$$

$$x_{p+2}(t+1) = \begin{cases} x_{p+2}(t) + 1 & \text{falls Kunde reagiert} \\ x_{p+2}(t) & \text{falls Kunde reagiert nicht} \end{cases}.$$

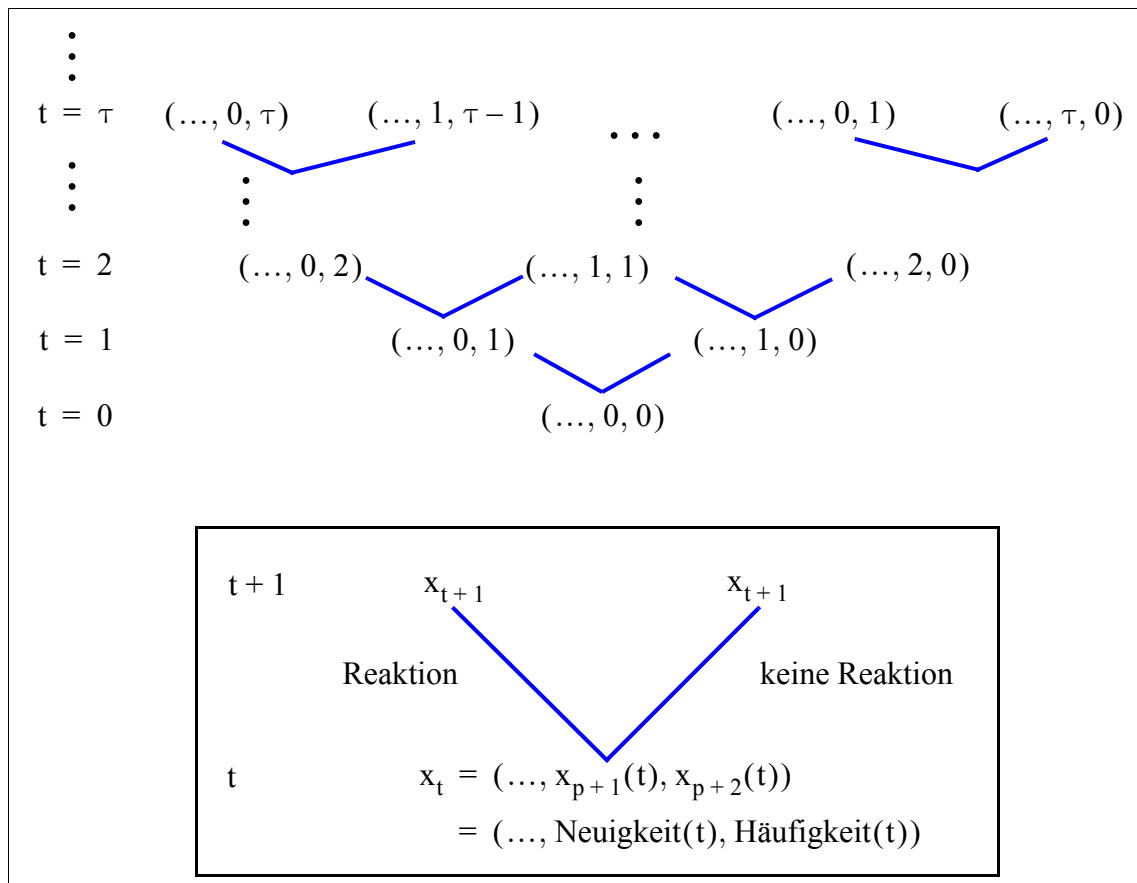


Abbildung 6.2: Projektion des Zustandsraumes in die Ebene, die von den Merkmalen „Neuigkeit“ und „Häufigkeit“ aufgespannt wird.

Alle Zustände σ_j mit $\xi_{p+1} = N_{p+1}$ oder $\xi_{p+2} = N_{p+2}$ sind Endzustände. Sie gehören entweder zu T_{pos} oder T_{neg} , so dass $\{\sigma_j \mid (\xi_{p+1} = N_{p+1}) \vee (\xi_{p+2} = N_{p+2})\} \subset T$. Für die entsprechenden Übergangswahrscheinlichkeiten gilt dann:

$$\forall \sigma_i \mid \xi_{p+1} = N_{p+1} - 1 : \sum_{\sigma_j \in T} p(\sigma_j \mid \sigma_i, \alpha) = 1 ,$$

$$\forall \sigma_i \mid \xi_{p+2} = N_{p+2} - 1 : \sum_{\sigma_j \in T} p(\sigma_j \mid \sigma_i, \alpha) = 1 .$$

Wenn das Merkmal „Häufigkeit“ als Quotient, wie in (5.4) definiert wird, dann kann die Erreichbarkeit eines Endzustands durch Einführung einer zusätzlichen Merkmalsvariable ξ_{p+3} gewährleistet werden. Sie gibt die Anzahl der Zustandsübergänge seit Beginn der Akquisition an, die mit der Anzahl der Akquiseaktionen übereinstimmt. Ihre Werte stammen aus der Merkmalsmenge $S_{p+3} = \{0, 1, \dots, N_{p+3}\}$. Weil die Anzahl der Zustandsübergänge seit Akquisitionsbeginn bekannt ist, wird der per Definition zu jeder Merkmalsmenge gehörende Wert „unbekannt“ nie angenommen. Zum Beginn der Akquisition ist der Wert des Merkmals stets gleich null. Der höchstmögliche Wert N_{p+3} wird vom Akquisiteur unter praktischen Gesichtspunkten festgelegt. Aus Erfahrung ist gewöhnlich bekannt, dass nach einer bestimmten Anzahl von Marketing-Aktionen (die der Anzahl von Zustandsübergängen entspricht) weitere Aktionen sinnlos sind und der Wert des Kundenmerkmals ξ_p auf den Wert „negKE“ gesetzt werden kann. Entsprechend sind alle Zustände σ_j mit $\xi_{p+3} = N_{p+3}$ Endzustände. Sie gehören entweder zu T_{pos} oder T_{neg} , so dass $\{\sigma_j \mid \xi_{p+3} = N_{p+3}\} \subset T$. Für die entsprechenden Übergangswahrscheinlichkeiten gilt dann:

$$\forall \sigma_i \mid \xi_{p+3} = N_{p+3} - 1 : \sum_{\sigma_j \in T} p(\sigma_j \mid \sigma_i, \alpha) = 1 .$$

6.5.3.2 Mehrfache Anwendung gleicher Aktionen

Ein weiteres Problem entsteht daraus, dass in der Praxis in der Regel die Anzahl der Zustände größer ist als die Anzahl der Aktionen, das heißt

$$|X| > |U| .$$

Das bedeutet, dass eine Politik dieselbe Aktion auf mehrere Zustände anwenden wird. Dies kann unerwünscht oder sogar kontraproduktiv sein, wenn diese Zustände hintereinander auftreten können. Solche Fälle können aber durch geeignete Festlegung der Übergangswahrscheinlichkeiten $p(\sigma_j \mid \sigma_i, \alpha)$, $\sigma_i, \sigma_j \in X$, $\alpha \in U$ vermieden werden.

6.5.3.3 Granularität der Merkmalsvariablen

Die fest vorgegebene Granularität der Merkmalsvariablen in Verbindung mit den Übergangswahrscheinlichkeiten ist nicht unproblematisch. Bei erstmaliger Durchführung einer Akquise werden die Übergangswahrscheinlichkeiten bei relativ feiner Granularität nur sehr ungenau

angegeben werden können. Dagegen wären Erfahrungswerte oder Daumenregeln, auf die in solchen Situationen zurückgegriffen werden muss, vertrauenswürdiger, wenn die Merkmalsvariable eher grobe Granularität besäßen. Auch bei der Gewinnung der Übergangswahrscheinlichkeiten durch statistische Auswertungen des beobachteten Kundenverhaltens, wäre bei empirischen Daten geringen Umfangs die Vertrauenswürdigkeit bei grober Granularität größer als bei einer feinen Granularität. Daraus wird ersichtlich,

- dass in der Praxis die Granularität von der Güte der Übergangswahrscheinlichkeiten abhängen sollte,
- dass die Granularität der Merkmalsvariablen flexibel sein sollte und
- Berechnungen der Erreichbarkeit eines Endzustandes und von optimalen Politiken bei jeder Änderung der Granularität erneut durchgeführt werden müssen, wobei ein Verfahren zu bevorzugen wäre, das dabei auf die bereits erzielten Ergebnisse zurückgreift.

In diesem Zusammenhang sind Ansätze interessant, die die Partitionierung in die Optimierung mit einbeziehen (z. B. in [131]).

6.6 Fallbeispiele: Dynamische Akquisepläne

In diesem Abschnitt wird gezeigt, dass die Hauptaussagen des Kapitels praktisch umsetzbar sind. Dazu werden Fallbeispiele betrachtet, die auf zwei dynamischen Akquiseplänen aufbauen. Der erste Akquiseplan ist für die praktische Anwendung zu einfach, dafür aber überschaubar und leicht interpretierbar. Der zweite Akquiseplan würde sich für die praktische Anwendung eignen. Er ist jedoch schwerer zu überschauen und zu interpretieren. Jeder Akquiseplan ist in einer *Excel*-Datei³⁹⁾ dokumentiert, die aus mehreren Tabellen besteht. Sie können hier wegen ihrer Größe nur auszugsweise wiedergegeben werden. Alle Einträge sind in englischer Sprache. Die vollständigen Dateien können von der Homepage des Verfassers⁴⁰⁾ heruntergeladen werden. Bezüge auf die Dateien erfolgen über die Dateinamen

- `Beispiel_1.xls` für den einfachen dynamischen Akquiseplan,
- `Beispiel_2.xls` für den komplexen dynamischen Akquiseplan

und die Tabellennamen.

Im Folgenden werden zunächst die beiden Akquisepläne vorgestellt (Abschnitte 6.6.1 und 6.6.2). Beide sind auf den Vertrieb einer *Hausratsversicherung* ausgelegt. Der zweite, komplexe Akquiseplan kann nicht vollständig manuell entworfen werden. Die softwaregestützte Erzeugung seines unscharfen Zieles, seiner unscharfen Nebenbedingungen und seiner stochastischen Matrizen wird in den Abschnitten 6.6.2.1 – 6.6.2.3 behandelt. Danach behandelt Abschnitt 6.6.3 die Implementierung der Algorithmen zur Überprüfung einer Politik auf Zulässigkeit (Abschnitt 6.6.3.2), zur Berechnung der unscharfen Entscheidung einer Politik (Abschnitt 6.6.3.3) und zur Berechnung einer optimalen Politik (Abschnitt 6.6.3.4). Beispiele für die Anwendung der Programme auf Grundlage der beiden Akquisepläne werden in den Abschnitten 6.6.4 – 6.6.5 gegeben. Die Berechnung einer optimalen Politik wird unter den

39. siehe Fußnote 40 auf Seite 186

40. <http://vieta.math.tu-cottbus.de/~klweber>

Gesichtspunkten Laufzeitverhalten (Abschnitt 6.6.5.1), Konvergenzverhalten (Abschnitt 6.6.5.2), Abhängigkeit der Lösung vom Startwert (Abschnitt 6.6.5.3) und von der verwendeten t-Norm (Abschnitt 6.6.5.4) näher untersucht. Eine fachliche Interpretation der berechneten optimalen Politiken wird im Abschnitt 6.6.5.5 vorgenommen.

6.6.1 Aufbau eines einfachen dynamischen Akquiseplans

Gemäß Def. 5.9 ist ein *dynamischer Akquiseplan* durch ein Tupel (X, T, U, P, C, G, π) gegeben, dessen Bestandteile nachfolgend für einen *einfachen* dynamischen Akquiseplan beschrieben sind.

1. Die *Zustandsmenge* X ist durch die Merkmalsvariablen und die Merkmalsmengen festgelegt. Dies sind:

- *Alter* mit den Werten „jung“, „alt“ und dem Standardwert „unbekannt“;
- *Bildungsstand* mit den Werten „gering“, „hoch“ und dem Standardwert „unbekannt“;
- *Produktinteresse* mit den Werten „interessiert“, „negative Kaufentscheidung“, „positive Kaufentscheidung“ und dem Standardwert „unbekannt“.

Der Charakter des einfachen dynamischen Akquiseplans als „Spielproblem“ wird an der groben Partitionierung der Merkmale deutlich. Aus den Kombinationen der Merkmalswerte ergeben sich die $3 \cdot 3 \cdot 4 = 36$ Zustände der Zustandsmenge $X = \{\sigma_1, \dots, \sigma_{36}\}$. Abb. 6.3 zeigt die Zustände mit ihren Merkmalswerten.

State	Age	Education	Interest				
1	unknown	unknown	unknown	19	unknown	unknown	negative
2	unknown	low	unknown	20	unknown	low	negative
3	unknown	high	unknown	21	unknown	high	negative
4	young	unknown	unknown	22	young	unknown	negative
5	young	low	unknown	23	young	low	negative
6	young	high	unknown	24	young	high	negative
7	old	unknown	unknown	25	old	unknown	negative
8	old	low	unknown	26	old	low	negative
9	old	high	unknown	27	old	high	negative
10	unknown	unknown	interested	28	unknown	unknown	positive
11	unknown	low	interested	29	unknown	low	positive
12	unknown	high	interested	30	unknown	high	positive
13	young	unknown	interested	31	young	unknown	positive
14	young	low	interested	32	young	low	positive
15	young	high	interested	33	young	high	positive
16	old	unknown	interested	34	old	unknown	positive
17	old	low	interested	35	old	low	positive
18	old	high	interested	36	old	high	positive

Abbildung 6.3: Auszug aus *Beispiel_1.xls*, Tabelle P1: Merkmalswerte der nicht beendeten Zustände 1 – 18 (links) und der beendeten Zustände 19 – 36 (rechts, blass).

2. Von den 36 Zuständen bildet die Gesamtheit der Zustände mit Produktinteresse: „negative Kaufentscheidung“ und „positive Kaufentscheidung“ die *Menge der beendeten Zustände*

$T = \{\sigma_{17}, \dots, \sigma_{36}\}$. Alle übrigen Zustände gehören zur Menge der nicht beendeten Zustände $\bar{T} = \{\sigma_1, \dots, \sigma_{16}\}$ (siehe Abb. 6.3).

3. Die Menge der *Akquise-Entscheidungen* U besitzt die vier Elemente

- α_1 : *Massenbrief* mit Antwortkupon zur Anforderung weiterer Informationen,
- α_2 : *Telefongespräch* bei dem die Werte aller Kundenmerkmale herausgefunden werden,
- α_3 : Brief mit einem *Informationsblatt* und Antwortkupon mit zwei Reaktionsmöglichkeiten für den Kunden: entweder Anforderung eines konkreten Angebots (was die Angabe des Alters erfordert) oder Bitte um einen Hausbesuch (ohne nähere Angaben),
- α_4 : Zusendung eines *Angebots* mit der Bitte um Bestätigung.

Wegen $|\bar{T}| = 18$ und $|U| = 4$ besitzt die Menge der Politiken insgesamt

$$|\Pi| = |U|^{|\bar{T}|} = 4^{18} = 68719476736$$

Politiken.

4. Die Menge der *stochastischen Übergangsmatrizen* P enthält für jede Akquise-Entscheidung eine Matrix: $P = \{P(\alpha_1), P(\alpha_2), P(\alpha_3), P(\alpha_4)\}$. Die Matrizen aus $[0, 1]^{|\bar{T}| \times |X|}$

besitzen für jeden nicht beendeten Zustand eine Zeile und für jeden beendeten Zustand eine Spalte. Die $|\bar{T}| \cdot |X| = 18 \cdot 36 = 648$ Wahrscheinlichkeitswerte für jede Matrix wurden von ABM-Kräften des Lehrstuhls für numerische und angewandte Mathematik an der Brandenburgischen Technischen Universität Cottbus aus der Erfahrung geschätzt und eingegeben. Bei der Größenordnung des einfachen dynamischen Akquiseplans war das problemlos in kurzer Zeit möglich, zumal viele Werte null sind.

5. Für jeden nicht beendeten Zustand wird – ebenfalls auf Erfahrungsgrundlage – eine *unscharfe Nebenbedingung* $C(x)$, $x \in \bar{T}$ über der Menge der vier Akquise-Entscheidungen U festgelegt. Die Zugehörigkeitswerte $\mu(\alpha_i | \sigma_j)$ geben an, wie angemessen die Akquise-Entscheidung α_i , $i = 1, \dots, 4$ im Zustand σ_j , $j = 1, \dots, 18$ ist. Die konkreten Mengen zeigt Abb. 6.5. Die Zugehörigkeitswerte der unscharfen Nebenbedingung $C(\sigma_7)$ sind blau eingerahmt. Über einen Kunden, der durch diesen Zustand beschrieben wird, ist nur bekannt, dass er *alt* ist („Age = old“). Die Zugehörigkeitswerte können wie folgt interpretiert werden:

- Ein *Massenbrief*, α_1 ist eine völlig angemessene Akquise-Entscheidung und wird deshalb mit Zugehörigkeitswert $\mu(\alpha_1 | \sigma_7) = 1$ bewertet. Durch den beiliegenden Antwortkupon ist es möglich, das Produktinteresse des Kunden zu erfahren und dann weitergehende Akquiseaktionen durchzuführen.
- Die Zusendung eines *Angebots* mit der Bitte um Bestätigung, α_4 wäre sehr unangemessen, weil über den Kunden zu wenig bekannt ist. Insbesondere ist sein Produktinteresse unbekannt. Der Aufwand eines Angebots ist in diesem Zustand nicht gerechtfertigt. Mög-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36			
1	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0.1	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0		
10	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0.2	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0.3	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0.3	0	

Abbildung 6.4: Auszug aus *Beispiel_1.xls*, Tabelle P3: Übergangswahrscheinlichkeiten für die Akquise-Entscheidung α_3 : „Brief mit einem *Informationsblatt*“ (siehe 3.) und die Zustände in Abb. 6.3. Es müssen nur die Übergangswahrscheinlichkeiten für die Übergänge von nicht beendeten Zuständen ($\sigma_1 - \sigma_{18}$) angegeben werden. Alle anderen Übergangswahrscheinlichkeiten sind entweder null oder eins.

State	Age	Education	Interest	α_1	α_2	α_3	α_4
1	unknown	unknown	unknown	1	0,5	0,2	0,1
2	unknown	low	unknown	1	0,6	0,3	0,1
3	unknown	high	unknown	1	0,4	0,3	0,1
4	young	unknown	unknown	1	0,4	0,3	0,1
5	young	low	unknown	0,7	0,6	0,3	0,1
6	young	high	unknown	0,7	0,4	0,3	0,1
7	old	unknown	unknown	1	0,3	0,3	0,1
8	old	low	unknown	0,7	0,5	0,3	0,1
9	old	high	unknown	0,7	0,3	0,3	0,1
10	unknown	unknown	interested	0,6	0,7	0,3	0,7
11	unknown	low	interested	0,6	0,7	0,3	0,4
12	unknown	high	interested	0,6	0,7	0,4	0,3
13	young	unknown	interested	0,6	0,7	0,3	0,3
14	young	low	interested	0,1	0,4	0,8	1
15	young	high	interested	0,1	0,4	0,8	1
16	old	unknown	interested	0,6	0,7	0,4	0,3
17	old	low	interested	0,1	0,4	0,8	1
18	old	high	interested	0,1	0,4	0,8	1

Abbildung 6.5: Auszug aus *Beispiel_1.xls*, Tabelle C: unscharfe Nebenbedingungen. Zu den nicht beendeten Zuständen auf der linken Seite sind rechts die Zugehörigkeitswerte der unscharfen Nebenbedingungen angegeben.

licherweise würde der Kunde die Zusendung eines Angebots als aufdringlich empfinden. Deshalb ist der Zugehörigkeitswert sehr klein: $\mu(\alpha_4 | \sigma_7) = 0,1$.

- Wegen des unbekannten Produktinteresses wären auch ein *Telefongespräch* oder die Zusendung eines *Informationsblattes* eher uneffektiv, und der Aufwand würde nicht dem möglichen Nutzen entsprechen. Die Akquise-Entscheidungen α_2 und α_3 wären jedoch

nicht so unangemessen wie α_4 und werden deshalb mit einem besseren, wenn auch geringen Zugehörigkeitswert $\mu(\alpha_2 | \sigma_7) = \mu(\alpha_3 | \sigma_7) = 0,3$.

6. Jeder beendete Zustand wird mit dem *unscharfen Ziel* bewertet (siehe Abb. 6.6). Die End-

State	Age	Education	Interest	G
19	unknown	unknown	negative	0
20	unknown	low	negative	0
21	unknown	high	negative	0
22	young	unknown	negative	0
23	young	low	negative	0
24	young	high	negative	0
25	old	unknown	negative	0
26	old	low	negative	0
27	old	high	negative	0
28	unknown	unknown	positive	0,5
29	unknown	low	positive	0,7
30	unknown	high	positive	0,7
31	young	unknown	positive	0,6
32	young	low	positive	1
33	young	high	positive	1
34	old	unknown	positive	0,6
35	old	low	positive	1
36	old	high	positive	1

Abbildung 6.6: Auszug aus `Beispiel_1.xls`, Tabelle G: Unscharfes Ziel; zu den beendeten Zuständen auf der linken Seite sind rechts (in rot) die Zugehörigkeitswerte des unscharfen Zieles angegeben.

zustände mit negativer Kaufentscheidung erhalten den Zugehörigkeitswert null: $\mu_G(\sigma_i)$, $i = 19, \dots, 27$. Mit Zugehörigkeitswert Eins erhalten die Zustände σ_{32} , σ_{33} , σ_{35} und σ_{36} die höchste Bewertung, weil alle Merkmale bekannt sind und so ein maßgeschneidertes – und damit (perspektivisch) den Gewinn maximierendes – Produkt verkauft werden kann.

7. Jede *Akquise-Entscheidungsregel* $\pi : X \rightarrow U$ wird als Tupel von Wertepaaren $(\sigma_i, \pi(\sigma_i))$, $i = 1, \dots, 18$ beschrieben. Ein Beispiel zeigt Abb. 6.7. Die Politik wurde unter folgenden Gesichtspunkten festgelegt:

- Wenn das Produktinteresse unbekannt ist und von den beiden anderen Merkmalen höchstens eines bekannt ist (Zustände $\sigma_1, \dots, \sigma_4, \sigma_7$), dann soll nur ein *Massenbrief* geschickt werden.
- Wenn das Produktinteresse unbekannt, aber die Werte der anderen Merkmale bekannt sind (Zustände $\sigma_5, \sigma_6, \sigma_8, \sigma_9$), dann soll ein *Telefonanruf* erfolgen.
- Wenn der Kunde am Produkt interessiert und jung ist (Zustände $\sigma_{13}, \dots, \sigma_{15}$) oder wenn er interessiert und die Werte aller anderen Merkmale bekannt sind, dann soll er ein *Angebot* erhalten.
- In allen anderen Fällen, soll ein Brief mit einem *Informationsblatt* zugesandt werden.

State	Age	Education	Interest	state σ	action α
1	unknown	unknown	unknown	1	1
2	unknown	low	unknown	2	1
3	unknown	high	unknown	3	1
4	young	unknown	unknown	4	1
5	young	low	unknown	5	2
6	young	high	unknown	6	2
7	old	unknown	unknown	7	1
8	old	low	unknown	8	2
9	old	high	unknown	9	2
10	unknown	unknown	interested	10	3
11	unknown	low	interested	11	3
12	unknown	high	interested	12	3
13	young	unknown	interested	13	4
14	young	low	interested	14	4
15	young	high	interested	15	4
16	old	unknown	interested	16	3
17	old	low	interested	17	4
18	old	high	interested	18	4

Abbildung 6.7: Auszug aus `Beispiel_1.xls`, Tabellen `P1` (links) und `policy_check` (rechts). Die Politik π ist durch die Wertepaare „state/action“, $(\sigma_i, \pi(\sigma_i))$, $i = 1, \dots, 18$ festgelegt.

Auf die Fragen, ob diese Politik zulässig ist, wie ihre unscharfe Entscheidung μ_D aussieht oder ob es eine bessere, eine optimale Politik gibt wird in den Abschnitten 6.6.4 – 6.6.5 eingegangen.

6.6.2 Aufbau eines komplexen dynamischen Akquiseplans

Wegen seiner Größe werden die Punkte 1. – 7. des vorigen Abschnitts hier nur kurz behandelt. Dabei wird deutlich, dass ein komplexer und damit wirklichkeitsnaher Akquiseplan nicht mehr manuell festgelegt werden kann. Wie das mit Software-Unterstützung möglich ist, wird in den Unterabschnitten gezeigt.

1. Die *Zustandsmenge* ist durch folgende Merkmalsvariablen mit den aufgeführten Merkmalswerten festgelegt:
 - *Produktinteresse* mit den Werten „interessiert“, „negative Kaufentscheidung“, „positiver Kaufentscheidung“ und dem Standardwert „unbekannt“;
 - *Alter* mit den Werten „jung“, „mittleren Alters“, „alt“ und dem Standardwert „unbekannt“;
 - *Bildungsstand* mit den Werten „ohne Berufsausbildung“ (z. B. ungelernt, angelernt), „Facharbeiter“, „akademische Ausbildung“ und dem Standardwert „unbekannt“;
 - *Kinder vorhanden* mit den Werten „ja“, „nein“ und dem Standardwert „unbekannt“.

Aus den Kombinationen der Merkmalswerte ergeben sich die $4 \cdot 4 \cdot 4 \cdot 3 = 192$ Zustände der Zustandsmenge $X = \{\sigma_1, \dots, \sigma_{192}\}$. Einige Zustände und ihre Merkmalswerte werden in Abb. 6.8 gezeigt.

State	Interest	Age	Education	Children					
66	interested	young	unskilled	yes	175	positive	middle-aged	skilled	unknown
67	interested	young	skilled	unknown	176	positive	middle-aged	skilled	no
68	interested	young	skilled	no	177	positive	middle-aged	skilled	yes
69	interested	young	skilled	yes	178	positive	middle-aged	academic	unknown
70	interested	young	academic	unknown	179	positive	middle-aged	academic	no
71	interested	young	academic	no	180	positive	middle-aged	academic	yes
72	interested	young	academic	yes	181	positive	old	unknown	unknown
73	interested	middle-aged	unknown	unknown	182	positive	old	unknown	no
74	interested	middle-aged	unknown	no	183	positive	old	unknown	yes
75	interested	middle-aged	unknown	yes	184	positive	old	unskilled	unknown
76	interested	middle-aged	unskilled	unknown	185	positive	old	unskilled	no
77	interested	middle-aged	unskilled	no	186	positive	old	unskilled	yes
78	interested	middle-aged	unskilled	yes	187	positive	old	skilled	unknown
79	interested	middle-aged	skilled	unknown	188	positive	old	skilled	no
80	interested	middle-aged	skilled	no	189	positive	old	skilled	yes
81	interested	middle-aged	skilled	yes	190	positive	old	academic	unknown
82	interested	middle-aged	academic	unknown	191	positive	old	academic	no
83	interested	middle-aged	academic	no	192	positive	old	academic	yes

Abbildung 6.8: Auszug aus `Beispiel_2.xls`, Tabelle P1: Merkmalswerte der nicht beendeten Zustände 66 – 83 (links) und der beendeten Zustände 175 – 182 (rechts, blass).

2. Von den 192 Zuständen bildet die Gesamtheit der Zustände mit Produktinteresse: „negative Kaufentscheidung“ und „positive Kaufentscheidung“ die *Menge der beendeten Zustände* $T = \{\sigma_{93}, \dots, \sigma_{192}\}$. Alle übrigen Zustände gehören zur Menge der nicht beendeten Zustände $\bar{T} = \{\sigma_1, \dots, \sigma_{92}\}$ (siehe Abb. 6.8).

3. Die Menge der *Akquise-Entscheidungen* U besitzt die sechs Elemente

- α_1 : *Massenbrief* mit Antwortkupon zur Anforderung weiterer Informationen,
- α_2 : *persönlicher Brief* mit Antwortkupon zur Anforderung einer telefonischen Beratung,
- α_3 : *Telefongespräch* bei dem die Werte aller Kundenmerkmale herausgefunden werden,
- α_4 : Brief mit einem *Informationsblatt* und Antwortkupon mit zwei Reaktionsmöglichkeiten für den Kunden: entweder Anforderung eines konkreten Angebots (was die Angabe des Alters erfordert) oder Bitte um einen Hausbesuch (ohne nähere Angaben),
- α_5 : *Hausbesuch* bei dem die Werte aller Kundenmerkmale herausgefunden werden,
- α_6 : Zusendung eines *Angebots* mit der Bitte um Bestätigung.

Wegen $|\bar{T}| = 96$ und $|U| = 6$ besitzt die Menge der Politiken insgesamt

$$|\Pi| = |U|^{|\bar{T}|} = 6^{96} = 5,041 \cdot 10^{74}$$

Politiken.

4. Die Menge der *stochastischen Übergangsmatrizen* P enthält für jede Akquise-Entscheidung eine Matrix: $P = \{P(\alpha_1), P(\alpha_2), P(\alpha_3), P(\alpha_4), P(\alpha_5), P(\alpha_6)\}$. Die Matrizen aus $[0, 1]^{|\bar{T}| \times |X|}$ besitzen für jeden nicht beendeten Zustand eine Zeile und für jeden beendeten

Zustand eine Spalte. Die $|\bar{T}| \cdot |X| = 96 \cdot 192 = 18432$ Wahrscheinlichkeitswerte für jede Matrix können nicht mehr manuell eingegeben werden. Zum einen ist der reine Arbeitsaufwand für die Eingabe unverhältnismäßig groß. Zum anderen ist die vorliegende Anzahl an Merkmalsvariablen, der Umfang der Merkmalsmengen und die Anzahl der Akquise-Entscheidungen so groß, dass auch erfahrene Experten die Übersicht verlieren und – trotz zahlreicher Null-Wahrscheinlichkeiten – ein vernünftiges Schätzen der Wahrscheinlichkeitswerte nicht mehr möglich ist. Eine Software-unterstützte Lösung des Problems wird im Abschnitt 6.6.2.1 beschrieben.

5. Für jeden nicht beendeten Zustand wird Software-unterstützt (siehe Abschnitt 6.6.2.2) eine *unscharfe Nebenbedingung* $C(x)$, $x \in \bar{T}$ über der Menge der sechs Akquise-Entscheidungen U festgelegt. Die Zugehörigkeitswerte $\mu(\alpha_i | \sigma_j)$ geben an, wie angemessen die Akquise-Entscheidung α_i , $i = 1, \dots, 6$ im Zustand σ_j , $j = 1, \dots, 96$ ist.
6. Jeder beendete Zustand mit dem unscharfen Ziel bewertet. Die Berechnung erfolgt ebenfalls Software-unterstützt (siehe Abschnitt 6.6.2.3).
7. Jede Akquise-Entscheidungsregel $\pi : X \rightarrow U$ wird als Tupel von Wertepaaren $(\sigma_i, \pi(\sigma_i))$, $i = 1, \dots, 96$ beschrieben.

6.6.2.1 Erzeugung der stochastischen Matrizen der Übergangswahrscheinlichkeiten

Das Problem bei der Schätzung von Übergangswahrscheinlichkeiten ist die Unübersichtlichkeit: Für jede der sechs Akquiseaktionen wären jeweils $96 \cdot 192$ Kombinationen aus Zustand und Folgezustand (jeweils gegeben durch vier Merkmalswerte) zu betrachten. Wenn geeignete Daten in ausreichendem Umfang vorhanden wären, könnten die Übergangswahrscheinlichkeiten daraus geschätzt werden. Die Voraussetzungen dafür sind jedoch nicht vorhanden und die dafür notwendigen Schätzverfahren sind ein eigenes Thema jenseits dieser Arbeit. Letztendlich muss den Wahrscheinlichkeiten also eine menschliche Einschätzung zugrunde liegen.

Laut Def. 5.2 ist die Zustandsmenge das kartesische Produkt der Merkmalsmenge

$$X = S_1 \times \dots \times S_p,$$

mithin jeder Zustand ein Tupel von Merkmalswerten $x = (\xi_1, \xi_2, \dots, \xi_p)$, $\xi_k \in S_k$, $k = 1, \dots, p$. Wird vereinfachend angenommen, dass für eine gegebene Akquiseaktion der Wert des Merkmals k im Zeitpunkt $t + 1$, $\xi_k^{(t+1)}$ nur vom Wert dieses Merkmals im Zeitpunkt t , $\xi_k^{(t)}$ abhängt, $k = 1, \dots, p$, dann ist

$$P(x_{t+1} | x_t, u_t) = \prod_{k=1}^p P(\xi_k^{(t+1)} | \xi_k^{(t)}, u_t) \quad (6.105)$$

für $t = 0, 1, \dots$. Mit dieser *Vereinfachung* braucht ein Experte nur die Übergangswahrscheinlichkeiten $P(\xi_k^{(t+1)} | \xi_k^{(t)}, u_t)$ für alle $|U| = 6$ Akquise-Entscheidungen und jeweils für alle

$p = 4$ Merkmale und jeden Merkmalswert von $\xi_k^{(t)}$, $k = 1, \dots, p$, der nicht alleine (das heißt unabhängig von den Werten der anderen Merkmalsvariablen) einen beendeten Zustand definiert, zu schätzen. Somit sind für die zwei Merkmalswerte „interessiert“ und „unbekannt“ des Merkmals *Produktinteresse* (nicht für die Merkmalswerte „negative Kaufentscheidung“, „positive Kaufentscheidung“, die eindeutig beendete Zustände definieren), für die jeweils vier Merkmalswerte der Merkmale *Alter* und *Bildungsstand*, für die drei Merkmalswerte des Merkmals *Kinder vorhanden* und jeweils für alle sechs Akquiseaktionen, also insgesamt $6 \cdot (2 + 4 + 4 + 3) = 78$ Übergangswahrscheinlichkeiten zu schätzen.

In der Praxis wird ein Experte (bewusst oder unbewusst) *Regeln* wie diese aufstellen:

Wenn *Produktinteresse* ist „unbekannt“ und *Akquiseaktion* ist Telefonanruf, **dann** wird das *Produktinteresse* (ach dem Telefonat

- in 30 % der Fälle: „unbekannt“,
- in 10 % der Fälle: „interessiert“,
- in 60 % der Fälle: „negative Kaufentscheidung“,
- in 0 % der Fälle: „positive Kaufentscheidung“

sein.

Hinter dieser Regel könnten z. B. folgende Überlegungen stehen:

- Ein erfahrener Versicherungsagent kann in einem Telefongespräch viele Einzelheiten über seinen Gesprächspartner herausfinden, unter anderem auch über dessen Produktinteresse (in 70 % der Fälle).
- Aus Erfahrung ist bekannt, dass etwa 60 % der Angerufenen das Versicherungsangebot entschieden ablehnen und nur 10 % interessiert sind.
- Die Erfahrung zeigt, dass aufgrund eines Telefongesprächs niemand eine positive Kaufentscheidung trifft (0 %).

Ein Instrument zur Formulierung und Verarbeitung sind Expertensystem-Werkzeuge, wie das hier verwendete *CLIPS*⁴¹⁾, in dessen Benutzerhandbuch folgender Zweck der Software genannt wird:

„CLIPS is designed to facilitate the development of software to model human knowledge or expertise.“ [60]

In der *CLIPS* eigenen Sprache werden Regeln, Fakten und ein Hauptprogramm in drei Textdateien bereitgestellt:

41. *CLIPS* wurde von der NASA entwickelt. Das erste Release wurde 1986 bereitgestellt. Die hier verwendete Version 6.21 der Software und Dokumentationen sind frei im Internet erhältlich (<http://www.ghg.net/clips/CLIPS.html>).

- `facts_TransProbs.txt` enthält Angaben zur Anzahl der Akquiseaktionen und dem Umfang der Merkmalsmengen sowie so genannte „templates“ zur Festlegung von Datenstrukturen.
- `rules_TransProbs.txt` enthält Regeln der oben genannten Art. In Abb. 6.9 sind die Regeln für das Merkmal *Alter* dargestellt. Weil es vier Merkmalswerte und sechs Akquiseaktionen gibt, könnten 24 Regeln aufgestellt werden. Nicht alle davon unterscheiden sich auf der rechten Seite. Diese können zusammengefasst werden, wie die Regeln `ageRule01`, `ageRule06`, `ageRule07` und `ageRule08` im gezeigten Beispiel. Wenn das Alter „unbekannt“ ist, dann wird sich durch einen Massenbrief („mailing“) oder einen persönlichen Brief mit Informationen („letter“) daran nichts ändern. Anstatt zwei Regeln aufzustellen, kann diese Tatsache durch eine Regel (`ageRule01`) ausgedrückt werden.
`rules_TransProbs.txt` enthält darüber hinaus eine Regel zur Kontrolle der Summe der Wahrscheinlichkeitswerte und zum Durchführen der Multiplikation in (6.105).
- `main.txt` enthält das Programm zur Steuerung des Programmablaufs: Laden von Regeln und Fakten, Iteration über alle Akquiseaktionen und über alle nicht beendeten Zustände, darin Aufruf der Inferenzmaschine und Ausgabe der Ergebnisse in eine Datei.

Aus Platzgründen werden hier nur Auszüge aus `rules_TransProbs.txt` gezeigt. Die vollständigen Dateien können von der Homepage des Verfassers⁴²⁾ heruntergeladen werden.

Die Regeln in `rules_TransProbs.txt` wurden vom Verfasser auf Grundlage des „gesunden Menschenverstands“ aufgestellt. Die Gedanken hinter jeder Regel sind jeweils als Kommentar (eingeleitet mit `;`) angegeben. Die Regeln entsprechen der Denkweise eines Experten, und deshalb konnten sie auch innerhalb weniger Stunden aufgestellt werden.

Die Datenflüsse sind in Abb. 6.10 illustriert: Der Experte gibt manuell die Regeln und Fakten zur Berechnung der Übergangswahrscheinlichkeiten in die Dateien `rules_TransProbs.txt` und `facts_TransProbs.txt` ein. Das Hauptprogramm `main.txt` wird mit *CLIPS* aufgerufen (`clipsdos.exe -f main.txt`). Es lädt die beiden Dateien mit Regeln und Fakten (angedeutet durch die gepunkteten Linien) und *CLIPS* erzeugt durch Abarbeitung der Regeln die sechs Dateien `transProbs_action1.txt`, ..., `transProbs_action6.txt`. Sie werden mit *Excel* in die Tabellen P1, ..., P6 der *Excel*-Datei `Beispiel_2.xls` importiert.

6.6.2.2 Erzeugung der unscharfen Nebenbedingungen

Für jeden nicht beendeten Zustand $\sigma_1, \dots, \sigma_{96}$ werden die sechs Akquise-Entscheidungen $\alpha_1, \dots, \alpha_6$ mit einer unscharfen Nebenbedingung $C(x) \in \text{IF}(U)$ bewertet. Das manuelle Schätzen der $96 \cdot 6 = 576$ Zugehörigkeitswerte $\mu(\alpha_i | \sigma_j)$, $i = 1, \dots, 6$, $j = 1, \dots, 96$ wäre aufwändig, aber nicht unmöglich. Vereinfacht wird die Aufgabe wie in Abschnitt 6.6.2.1 durch getrennte Betrachtung der Merkmale, deren kartesisches Produkt die Zustandsmenge ist. Zu diesem Zweck werden die unscharfen Mengen

42. <http://vieta.math.tu-cottbus.de/~klweber>

```

;; -----
;; Rules to map age values to newAge values according to given action;
;; NewAge values are pairs of attribute value and value probability.
;; -----

(defrule ageRule01
  (age unknown)
  (action mailing | letter)
=>
  ;; Reply to a mailing or letter using the attached coupon does not
  ;; give information on the age. Thus, knowledge on age does not change.
  (assert (newAge (unknown 1.0) (young 0.0) (middle-aged 0.0) (old 0.0)))
)

(defrule ageRule02
  (age unknown)
  (action phone)
=>
  ;; Assumption: An experienced insurance agent is able to estimate the
  ;; age of his counterpart on the phone from the voice. These persons
  ;; are distributed according to their presence in society.
  (assert (newAge (unknown 0.1) (young 0.25) (middle-aged 0.4) (old 0.25)))
)

(defrule ageRule03
  (age unknown)
  (action flyer)
=>
  ;; The flyer has a reply coupon to request an offer (which requires age
  ;; indication) or to request a phone call. According to "intNew04" and
  ;; "intNew10" 8 % to 15 % answer positively. Thus, here the mean value
  ;; is taken: 12 %. This portion is distributed among age values similar
  ;; to "ageRule02". 88 % do not reply at all.
  (assert (newAge (unknown 0.88) (young 0.034) (middle-aged 0.052) (old 0.034)))
)

(defrule ageRule04
  (age unknown)
  (action visit)
=>
  ;; This case is quite similar to "ageRule02", however from personal meeting
  ;; the experienced insurance agent can estimate age in any case. Age
  ;; distribution follows society.
  (assert (newAge (unknown 0.0) (young 0.275) (middle-aged 0.45) (old 0.275)))
)

(defrule ageRule05
  (age unknown)
  (action offer)
=>
  ;; Assumption: Either the offer is confirmed or not. If it is confirmed
  ;; the customer indicates his age. Age distribution follows society.
  ;; Offers are only sent to prospective customers. 30 % do not confirm.
  (assert (newAge (unknown 0.3) (young 0.2) (middle-aged 0.3) (old 0.2)))
)

(defrule ageRule06
  (age young)
  (action mailing | letter | phone | flyer | visit | offer)
=>
  ;; Assumption: Once the age is known, it does not change by any action.
  (assert (newAge (unknown 0.0) (young 1.0) (middle-aged 0.0) (old 0.0)))
)

(defrule ageRule07
  (age middle-aged)
  (action mailing | letter | phone | flyer | visit | offer)
=>
  ;; Assumption: Once the age is known, it does not change by any action.
  (assert (newAge (unknown 0.0) (young 0.0) (middle-aged 1.0) (old 0.0)))
)

(defrule ageRule08
  (age old)
  (action mailing | letter | phone | flyer | visit | offer)
=>
  ;; Assumption: Once the age is known, it does not change by any action.
  (assert (newAge (unknown 0.0) (young 0.0) (middle-aged 0.0) (old 1.0)))
)

```

Abbildung 6.9: Auszug aus `rules_TransProbs.txt`: Zu sehen sind die Regeln für das Merkmal *Alter* in der *CLIPS*-Sprache.

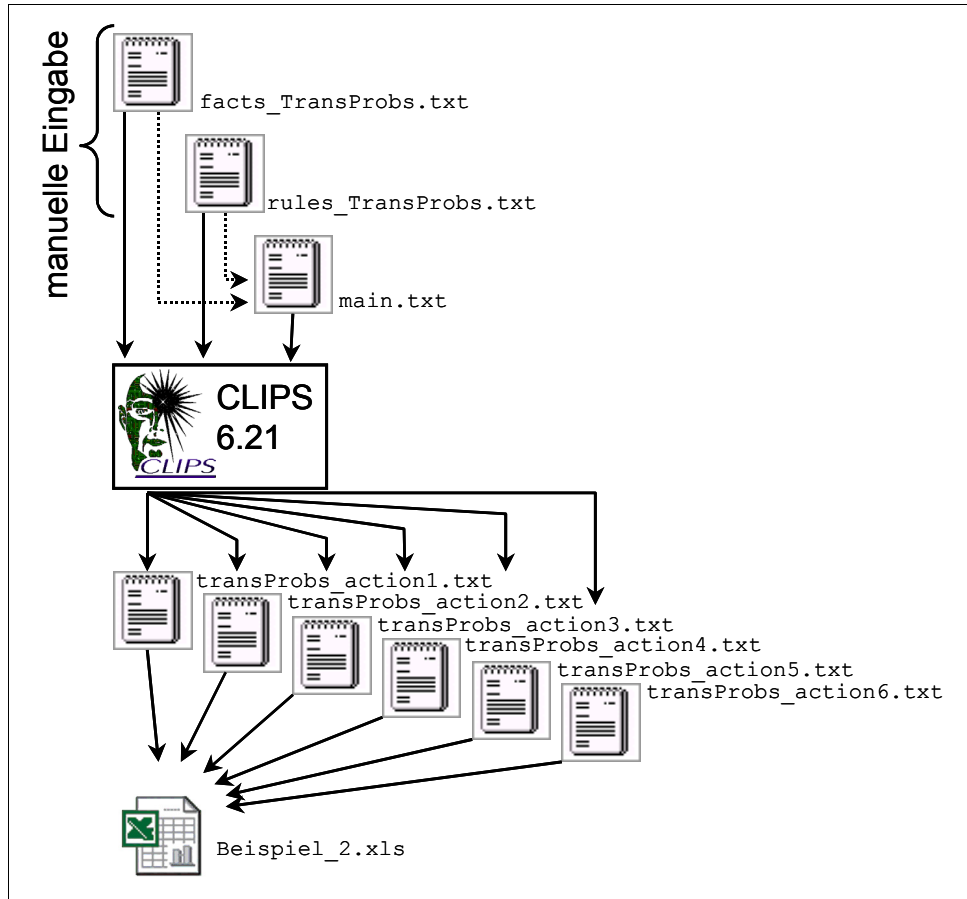


Abbildung 6.10: Schematische Darstellung der Datenflüsse (durchgezogene Pfeile) und der Abhängigkeiten im CLIPS-Programm (gestrichelte Pfeile).

$$C(\xi_i) = \{(u, \mu(u | \xi_i)) \mid u \in U\} \in \text{IF}(U), \xi_i \in S_i, i = 1, \dots, p$$

eingeführt sowie die vereinfachende Annahme getroffen, dass für $x = (\xi_1, \xi_2, \dots, \xi_p)$ gilt:

$$C(x) = C(\xi_1) \times_t \dots \times_t C(\xi_p). \quad (6.106)$$

Das kartesische Produkt der unscharfen Mengen $C(\xi_i) \in \text{IF}(U)$ in (6.106) ist nach Def. 2.11:

$$C(x) = C(\xi_1) \times_t \dots \times_t C(\xi_p) = \{(\xi_1, \xi_2, \dots, \xi_p) \parallel C(\xi_1) \wedge_t \dots \wedge_t C(\xi_p)\}.$$

Damit berechnen sich die Zugehörigkeitswerte von $C(x)$ wie folgt:

$$\mu(u | x) = \mu(u | \xi_1) \mathbf{t} \dots \mathbf{t} \mu(u | \xi_p). \quad (6.107)$$

In der Literatur und in den klassischen Arbeiten von Zadeh wird für die t-Norm die Minimumbildung verwendet. So wird auch hier verfahren. Das bedeutet, für die Erstellung der unscharfen Nebenbedingungen hat ein Experte für jeden Merkmalswert ξ_i (der nicht zu einem beendeten Zustand gehört) jedes Merkmals i , $i = 1, \dots, p$ und für alle Akquiseaktionen aus U

durch Festlegung eines Zugehörigkeitswertes $\mu(u | \xi_i)$ abzuschätzen, wie gut die jeweilige Akquiseaktion zu dem Merkmalswert passt.

In der Praxis wird ein Experte (bewusst oder unbewusst) *Regeln* wie diese aufstellen:

Wenn das Merkmal *Alter* den Merkmalswert $\xi = \text{„jung“}$ besitzt, dann sind die Akquiseaktionen

- α_1 , *Massenbrief* mit Antwortkupon zur Anforderung weiterer Informationen: eher passend: $\mu(u | \xi_i) = 0,6$;
- α_2 , *persönlicher Brief* mit Antwortkupon zur Anforderung einer telefonischen Beratung: eher unpassend: $\mu(u | \xi_i) = 0,4$;
- α_3 , *Telefongespräch* bei dem die Werte aller Kundenmerkmale herausgefunden werden: ideal: $\mu(u | \xi_i) = 1$;
- α_4 , Brief mit einem *Informationsblatt* und Antwortkupon: ideal: $\mu(u | \xi_i) = 1$;
- α_5 , *Hausbesuch* bei dem die Werte aller Kundenmerkmale herausgefunden werden: eher ungeeignet: $\mu(u | \xi_i) = 0,3$;
- α_6 , Zusendung eines *Angebots* mit der Bitte um Bestätigung: ideal: $\mu(u | \xi_i) = 1$.

In Abb. 6.11 werden diese Regel und alle anderen Regeln für das Merkmal *Alter* in der Sprache von *CLIPS* gezeigt. Dort stehen in Kommentaren (eingeleitet durch `;`) auch die Überlegungen des Verfassers hinter den Regeln. Durch die beschriebene Vereinfachung sind letztlich nicht 576 Zugehörigkeitswerte festzulegen, sondern nur für jeden Merkmalswert (ohne die Merkmalswerte „negative Kaufentscheidung“ und „positive Kaufentscheidung“, die eindeutig beendete Zustände definieren) sechs davon, also $(2 + 4 + 4 + 3) \cdot 6 = 78$. Wie bei der Erzeugung der Übergangswahrscheinlichkeiten im Abschnitt 6.6.2.1 werden Regeln, Fakten und ein Hauptprogramm bereitzustellen:

- `facts_Constraints.txt` enthält so genannte „templates“ zur Festlegung der Datenstruktur für die unscharfen Nebenbedingungen (`ageFuzzyConst` in Abb. 6.11).
- `rules_Constraints.txt` enthält Regeln der oben genannten Art sowie eine Regel zur „Aggregation“ der Zugehörigkeitswerte gemäß (6.107), wobei als t-Norm die Minimumbildung verwendet wird.
- `main.txt` ist die Erweiterung des Hauptprogramms aus Abschnitt 6.6.2.1 um das Laden von Regeln und Fakten aus `facts_Constraints.txt` und `rules_Constraints.txt` und Ausgabe der Ergebnisse in die Datei `constraints.txt`.

```

;; Rules which evaluate actions according to the
;; prospective customer's age.
;; -----

(defrule ageConstRule01
  (age unknown)
=>
  ;; Since the letter is age-specific, it is inappropriate to send one in this case.
  ;; A mailing can bring about more information. A phone call, a flyer, and a visit
  ;; can help to get more information. An offer is inappropriate, because without
  ;; knowledge of the age, it cannot be tailored to the prospective customer's needs.
  (assert (ageFuzzyConst (mailing 0.75) (letter 0.5) (phone 1.0) (flyer 1.0) (visit 1.0) (offer 0.25)) )
)

(defrule ageConstRule02
  (age young)
=>
  ;; Young people are always in a hurry and do not like classical means of marketing.
  ;; Many of them decide impulsively on base of first impression.
  (assert (ageFuzzyConst (mailing 0.6) (letter 0.4) (phone 1.0) (flyer 1.0) (visit 0.3) (offer 1.0)) )
)

(defrule ageConstRule03
  (age middle-aged)
=>
  ;; Middle-aged people are always in a hurry. They are rather serious and decide on base of
  ;; information. They like to be convinced but not to be persuaded. Thus, mailings and letters
  ;; are of no use. A phone call could be helpful. A visit could help or harm. A flyer or an offer
  ;; give all the information the prospective customer likes to know for his decision.
  (assert (ageFuzzyConst (mailing 0.0) (letter 0.2) (phone 0.6) (flyer 1.0) (visit 0.5) (offer 1.0)) )
)

(defrule ageConstRule04
  (age old)
=>
  ;; Elderly people have time to read printed marketing material and to discuss with the agent.
  ;; However, they do not like phone calls, or it may be difficult to discuss with them on the
  ;; phone.
  (assert (ageFuzzyConst (mailing 1.0) (letter 1.0) (phone 0.2) (flyer 1.0) (visit 1.0) (offer 1.0)) )
)

```

Abbildung 6.11: Auszug aus `rules_Constraints.txt`: Zu sehen sind die Regeln für das Merkmal *Alter* in der *CLIPS*-Sprache.

6.6.2.3 Erzeugung des unscharfen Zieles

Bei der Festlegung des unscharfen Zieles wird völlig analog zur Festlegung der unscharfen Nebenbedingungen verfahren. Die unscharfe Menge $G \in \text{IF}(X)$ ist eine unscharfe Menge über dem kartesischen Produkt $X = S_1 \times \dots \times S_p$. Anstatt nun für alle 96 nicht beendeten Zustände $\sigma_{97}, \dots, \sigma_{192}$ jeweils einen Zugehörigkeitswert $\mu_G(x)$ festzulegen, wird angenommen, dass G das unscharfe kartesische Produkt der unscharfen Mengen $G_i \in \text{IF}(S_i)$ für $i = 1, \dots, p$ ist und dass $G = G_1(\xi_1) \times_t \dots \times_t G_p(\xi_p)$ und mithin

$$\mu_G(x) = \mu_{G_1}(\xi_1) \mathbf{t} \dots \mathbf{t} \mu_{G_p}(\xi_p) \quad (6.108)$$

gelten. Damit sind nur noch für alle Merkmalswerte, die in beendeten Zuständen vorkommen können, Zugehörigkeitswerte zu schätzen, also für die Merkmalswerte „positive Kaufentscheidung“ und „negative Kaufentscheidung“ des Merkmals *Produktinteresse* und für alle Merkmalswerte der Merkmale *Alter*, *Bildungsstand*, *Kinder vorhanden*. Dies sind $2 + 4 + 4 + 3 = 13$ Zugehörigkeitswerte. Wie bei den unscharfen Nebenbedingungen erfolgt in *CLIPS* die Festlegung der Zugehörigkeitswerte über Regeln in der Datei `rules_Goal.txt`. Einen Auszug zeigt Abb. 6.12. Diese Datei enthält auch eine Regel zur Berechnung von (6.108) mit der t-Norm Minimumsbildung. Das Hauptprogramm `main.txt`

```

;; Rules which evaluate terminal states according to the
;; prospective customer's decision ("interest").
;; -----
(defrule intRule01
  (int negative)
=>
  ;; Negative interest mean no contract!
  (assert (intGoal 0.0))
)

(defrule intRule02
  (int positive)
=>
  ;; Positive interest means contract conclusion!
  (assert (intGoal 1.0))
)

;; Rules which evaluate terminal states according to the
;; prospective customer's age.
;; -----
(defrule ageRule01
  (age unknown)
=>
  ;; If age is unknown it is more likely that the offer
  ;; is not tailored and thus insurance sum is low.
  (assert (ageGoal 0.4))
)

(defrule ageRule02
  (age young)
=>
  ;; Young people usually do not have that much property than older
  ;; people. Normally, their insurance sums are below the average.
  (assert (ageGoal 0.3))
)

(defrule ageRule03
  (age middle-aged)
=>
  ;; Middle-aged people have much property, seek to get even more
  ;; and secure what they have. Therefore, the insurance sum is
  ;; normally high.
  (assert (ageGoal 1.0))
)

(defrule ageRule04
  (age old)
=>
  ;; Elderly people do not have the need for security than
  ;; middle-aged people. Therefore, their insurance sums are
  ;; rather smaller.
  (assert (ageGoal 0.75))
)

```

Abbildung 6.12: Auszug aus `rules_Goal.txt`: Zu sehen sind die Regeln für die Merkmale *Produktinteresse* und *Alter* in der *CLIPS*-Sprache.

wird entsprechend um das Laden der Regeln aus `rules_Goal.txt` erweitert. Eine vollständige Darstellung aller Datenflüsse (als Erweiterung von Abb. 6.10) zeigt Abb. 6.13. Die zusätzlichen Ausgabe-Dateien `constraints.txt` und `goal.txt` werden mit *Excel* in die Tabellen C und G der *Excel*-Datei `Beispiel_2.xls` importiert.

6.6.3 Implementierung der Verfahren

Die in diesem Kapitel hergeleiteten Verfahren

- zur Prüfung einer Politik auf Zulässigkeit,
- zur Berechnung der unscharfen Entscheidung einer zulässigen Politik,
- zur Berechnung einer optimalen Politik für einen dynamischen Akquisepfad

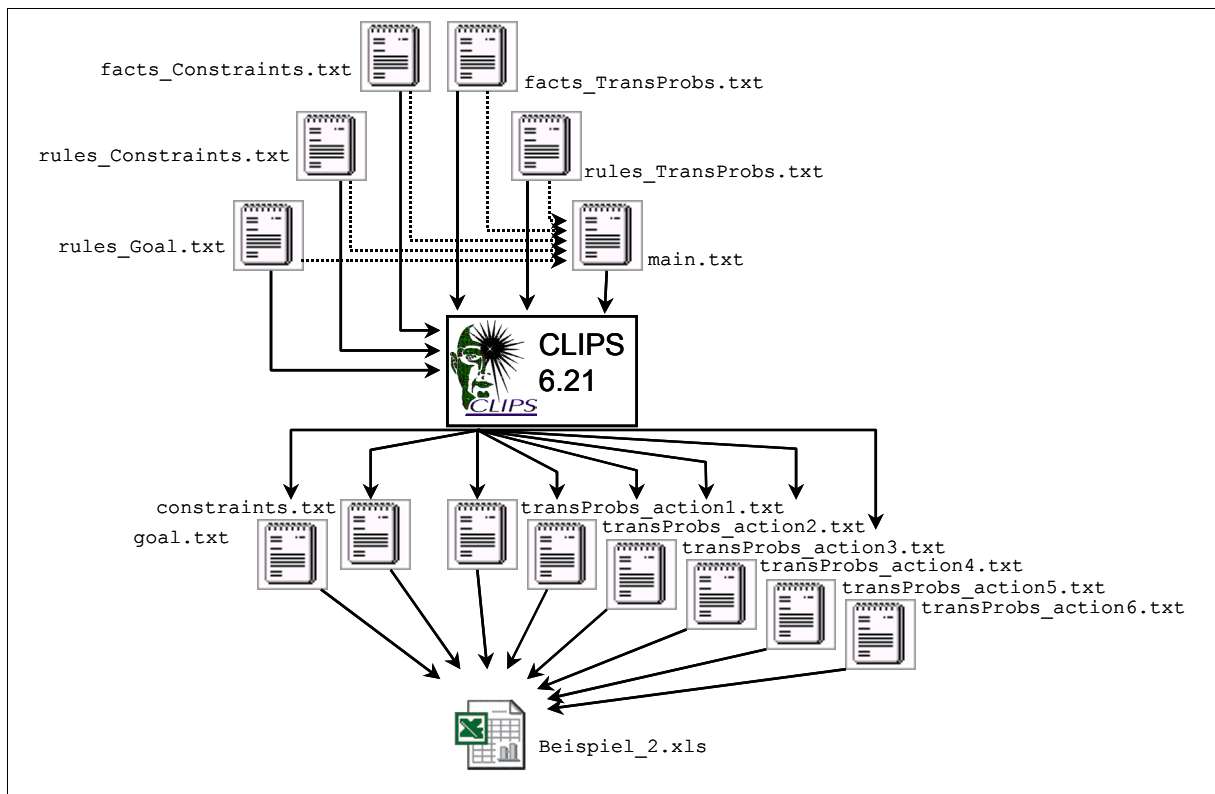


Abbildung 6.13: Schematische Darstellung der Datenflüsse (durchgezogene Pfeile) und der Abhängigkeiten im CLIPS-Programm (gestrichelte Pfeile).

wurden mit *MATLAB* implementiert⁴³⁾. In den folgenden Unterabschnitten wird der Programm-Code erläutert. Die vollständigen *MATLAB*-Dateien können von der Homepage des Verfassers⁴⁴⁾ herunter geladen werden. Alle Eingabedaten werden in den *Excel*-Dateien *Beispiel_1.xls* (einfacher dynamischer Akquiseplan, siehe Abschnitt 6.6.1) und *Beispiel_2.xls* (komplexer dynamischer Akquiseplan, siehe Abschnitt 6.6.2) bereitgestellt. Berechnungsergebnisse werden in Textdateien geschrieben, die mit *Excel* in die jeweilige *Excel*-Datei importiert werden, so dass dass sämtliche Datenein- und -ausgaben über die beiden *Excel*-Dateien erfolgen kann.

6.6.3.1 Einlesen und Ausgabe der Daten

Das Einlesen aller Eingabedateien erfolgt durch die Funktion `initialize` (siehe Alg. 6.1) in der Datei `initialize.m`. Sie besitzt folgende Syntax-Definition

```
function [parameter, PT_, PT, mueC, mueG, omega0, policy] =
    initialize(xlstabname)
```

Ihr **Eingabevariablen** ist:

43. *MATLAB* ist eine Programmiersprache für technische Berechnungen und eine dazugehörige Entwicklungsumgebung der Firma *The MathWorks* (<http://www.mathworks.de>). Verwendet wurde Version 6.5 Release 13.

44. <http://vieta.math.tu-cottbus.de/~klweber>

- **xlstabname:** Der Name der *Excel*-Datei (hier *Beispiel_1.xls* oder *Beispiel_2.xls*).

Ihre **Rückgabevariablen** sind:

- **parameter:** Eine Liste von Verfahrensparametern, die aus der Tabelle *Parameter* der *Excel*-Dateien eingelesen wird, die in Abb. 6.14 dargestellt sind. Soweit sie nicht selbster-

	Parameter	Value
1	number of states	192
2	number of non-terminated states	96
3	number of actions/decisions	6
4	ε -value: termination criterion	1,00E-12
5	max. number of iterations (termination criterion)	10000
6	flag for t-norm "minimum" (1: on, 0: off)	1
7	flag for t-norm "algebraic product" (1: on, 0: off)	1
8	flag for t-norm "bounded difference" (1: on, 0: off)	1
9	flag for t-norm "drastic product" (1: on, 0: off)	1
10	flag for t-norm "Hamacher product" (1: on, 0: off)	1
11	flag for t-norm "Einstein product" (1: on, 0: off)	1
12	mode of operation: 1: check feasibility of given policy, 2: check <parameter 13> randomly generated policies	1
13	number of policies to be generated randomly (in case parameter 12 is 2)	100

Abbildung 6.14: Tabelle *Parameter* in den *Excel*-Dateien *Beispiel_1.xls* mit Verfahrensparametern.

klärend sind, werden sie in den Unterabschnitten beim jeweiligen Verfahren erläutert.

- **PT₋, PT:** zwei Listen, die die Matrizen mit den Übergangswahrscheinlichkeiten für alle Akquiseaktionen enthalten, also: $P_{\bar{T}}(\alpha_1), \dots, P_{\bar{T}}(\alpha_{|U|})$ und $P_T(\alpha_1), \dots, P_T(\alpha_{|U|})$ (vgl. Def. 6.2 und (6.35)). Die Matrizen werden aus den Tabellen *P1*, *P2*, ... der *Excel*-Datei eingelesen.
- **mueC:** eine $|\bar{T}| \times |U|$ -Matrix mit den Zugehörigkeitswerten der unscharfen Nebenbedingungen: $\mu_C(u | x)$, $u \in U$ und $x \in \bar{T}$. Die Matrix wird aus der Tabelle *C* der *Excel*-Datei eingelesen.
- **mueG:** ein Vektor mit den Zugehörigkeitswerten des unscharfen Ziels: $\mu_G(x)$, $x \in T$. Der Vektor wird aus der Tabelle *G* der *Excel*-Datei eingelesen.
- **omega0:** Ein Initialisierungsvektor aus $[0, 1]^{|\bar{T}|}$ für die Transformationen (6.96) und (6.97), der aus Tabelle *omega0* der *Excel*-Datei eingelesen wird.
- **policy:** Ein Vektor aus $U^{|\bar{T}|}$, der eine Politik $\pi = (\pi(\sigma_1), \dots, \pi(\sigma_k))$, $k = |\bar{T}|$ definiert. Dieser Vektor kann auf Zulässigkeit untersucht werden. Er wird aus der Tabelle *policy_check* der *Excel*-Datei eingelesen.

Die Funktionen zur Ausgabe der Ergebnisse werden in den folgenden Abschnitten mit den Berechnungsverfahren erklärt. Grundsätzlich werden Ergebnisse auf dem Bildschirm im *MATLAB* „Command Window“ ausgegeben und in Textdateien geschrieben. Aus diesen Dateien werden sie mit *Excel* in Tabellen der *Excel*-Datei *Beispiel_1.xls* (einfacher dynamischer Akquiseplan) oder der *Excel*-Datei *Beispiel_2.xls* (komplexer dynamischer Akquiseplan) importiert.

Die optimalen Politiken (Verfahren siehe Abschnitt 6.6.3.3) für unterschiedliche t-Normen (siehe Def. 2.7) werden in die folgenden Textdateien ausgegeben (jeweils für $u, v \in [0, 1]$) und von dort mit *Excel* in die Tabelle *result_policy* importiert:

- *result_AlgProd.txt*: Ausgabe für t-Norm „algebraisches Produkt“ (6.90),
- *result_BdDiff.txt*: Ausgabe für t-Norm „beschränkte Differenz“ (6.92),
- *result_DrastProd.txt*: Ausgabe für t-Norm „drastisches Produkt“ (6.89),
- *result_EinstProd.txt*: Ausgabe für t-Norm „Einstein-Produkt“ (6.94),
- *result_HamProd*: Ausgabe für t-Norm „Hamacher-Produkt“ (6.93).
- *result_Min*: Ausgabe für t-Norm „Minimum“ (6.88).

Das Ergebnis der Überprüfung einer gegebenen Politik bzw. zufällig generierter Politiken auf *Zulässigkeit* (Verfahren siehe Abschnitt 6.6.3.2) wird für unterschiedliche t-Normen (siehe Def. 2.7) in die folgenden Textdateien ausgegeben:

- *result_CheckFeasGiven.txt*: Überprüfung einer vorgegebenen Politik, *Excel*-Import in die Tabelle *policy_check*;
- *result_CheckFeasRandom.txt*: Überprüfung einer oder mehrerer zufällig erzeugter Politiken, *Excel*-Import in die Tabelle *random_result*.

Die Berechnung der *unscharfen Entscheidung* (Verfahren siehe Abschnitt 6.6.3.3) zu einer zulässigen Politik wird für unterschiedliche t-Normen (siehe Def. 2.7) in die folgende Textdatei ausgegeben und mit *Excel*-Import in die Tabelle *policy_check* importiert:

- *result_fuzzyDecision*.

6.6.3.2 Verfahren zur Bestimmung der Zulässigkeit einer Politik

Die Grundlage des Verfahrens zur Bestimmung der Zulässigkeit einer Politik eines dynamischen Akquiseplans ist Satz 6.6 im Abschnitt 6.4.2. Es wird in *MATLAB* von der Funktion

```
function [isFeas, K, PT_pi_K] = checkGivenPolicy(nT_, nU,
PT_, policy)
```

in der Datei *checkGivenPolicy.m* realisiert. In Abb. 6.15 sind die wesentlichen Teile des *MATLAB*-Programmkodes wiedergegeben.

Die **Eingabevariablen** sind Elemente der Liste *parameter*, die von der Funktion *initialize* erzeugt wird (siehe Abschnitt 6.6.3.1):

```

% =====%
% Function to check feasibility of given policy %
% =====%

function [isFeas, K, PT_pi_K] = checkGivenPolicy(nT_, nU, PT_, policy)

...

% Computation of transition probability matrix for given policy
for i = 1:nT_
    PT_pi(i,:) = PT_{policy(i)}(i,:);
end

% Initialize feasibility flag
isFeas = 0;

% Compute powers of matrix PT_ until norm < 1
K = 0;
while ((isFeas ~= 1) & (K <= nT_))
    K = K + 1;
    PT_pi_K = PT_pi ^ K;
    normValue = norm(PT_pi_K, inf);
    if normValue < 1
        isFeas = 1;
    end;
end;
end;

```

Abbildung 6.15: Auszug aus dem *MATLAB*-Programmcode der Funktion `checkGivenPolicy` in der Datei `checkGivenPolicy.m`.

- $nT_$: die Anzahl der nicht beendeten Zustände $|\bar{T}|$;
- nU : die Anzahl der Entscheidungen $|U|$;
- $PT_$: siehe Seite 324;
- `policy`: siehe Seite 324.

Ihre **Rückgabevariablen** sind:

- `isFeas`: Kennzeichen („flag“), das angibt, ob die Politik `policy` zulässig ist (Wert 1) oder nicht (Werte 0);
- K : kleinste Potenz der Matrix $P_{\bar{T}}(\pi)$, gegeben durch $PT_$ und `policy`, für die gilt:

$$\|P_{\bar{T}}^K(\pi)\| < 1;$$
- PT_pi_K : Matrix $P_{\bar{T}}^K(\pi)$, gegeben durch $PT_$, `policy` und K .

Der Algorithmus ist im *MATLAB*-Programmcode in Abb. 6.15 ersichtlich. Wesentlich sind die letzten acht Zeilen des Programms. Es werden nacheinander Potenzen der Matrix $P_{\bar{T}}(\pi)$

berechnet: $P_T^K(\pi)$, $K = 1, 2, \dots$, solange bis entweder $\|P_T^K(\pi)\| < 1$ gilt, also die Politik *zulässig* ist, oder $K > k = |\bar{T}|$ und somit die Politik *nicht zulässig* ist.

Die Funktion `checkGivenPolicy` wird im *MATLAB*-Programm `checkFeasibility.m` aufgerufen. In Abhängigkeit des Parameters 12 (siehe Abb. 6.14) wird entweder eine über die Tabelle `policy_check` der Excel-Datei eingelesene Politik auf Zulässigkeit untersucht (Parameterwert 1) oder es werden zufällige Politiken erzeugt und untersucht (Parameterwert 2). Die Anzahl der zufällig erzeugten Politiken wird durch Parameter 13 bestimmt. Im zweiten Fall wird die Politik in der Tabelle `policy_check` nicht beachtet. Die Möglichkeit eine vorgegebenen Anzahl von Politiken zufällig zu erzeugen erlaubt die Untersuchung des Algorithmus anhand zahlreicher Politiken auch bei großen Zustandsräumen.

Mit der Funktion

```
function outCheckFeasibility(outFile, policy, isFeas, K)
```

können die Ergebnisse einer Zulässigkeitsuntersuchung mit dem Programm `checkFeasibility.m` in eine Datei ausgegeben werden. Die Funktion ist überladen⁴⁵). Die Eingabevariablen `policy`, `isFeas`, `K` stimmen entweder mit den Ausgabevariablen der Funktion `checkGivenPolicy` (siehe oben) überein, oder es handelt sich um Felder zur Aufnahme dieser Größen für mehrere Politiken – wie bei der Untersuchung mehrerer zufällig erzeugter Politiken. Im Programm `checkFeasibility.m` wird der Variablen `outFile` je nach Fall (Untersuchung einer vorgegebenen Politik oder Untersuchung mehrerer zufällig erzeugter Politiken, siehe oben) entweder der Wert `result_CheckFeasGiven.txt` oder der Wert `result_CheckFeasRandom.txt` (siehe Abschnitt 6.6.3.1) übergeben. Einen Überblick über die Funktionsaufrufe im *MATLAB*-Programm und die Datenflüsse zwischen Excel-Datei, *MATLAB*-Programm und Textdateien zeigt Abb. 6.16.

6.6.3.3 Verfahren zur Berechnung der unscharfen Entscheidung einer Politik

Die Grundlage des Verfahrens zur Berechnung der unscharfen Entscheidung einer Politik eines dynamischen Akquiseplans ist Satz 6.9 im Abschnitt 6.4.2. Nachdem im Abschnitt 6.6.3.2 der grundsätzliche Aufbau eines *MATLAB*-Programms mit Funktionsaufrufen sowie der Datenfluss zwischen *MATLAB*-Programm, Excel-Datei und Textdateien vorgestellt worden ist, geht die Beschreibung in diesem Abschnitt gleich von der schematischen Übersichtsdarstellung in Abb. 6.17 aus.

Die Berechnung der unscharfen Entscheidung zu einer gegebenen unscharfen Politik wird vom *MATLAB* Programm `fuzzyDecision` in der Datei `fuzzyDecision.m` durchgeführt. Es benutzt die im Abschnitt 6.6.3.1 vorgestellten Funktionen `initialize` zum Datenladen und

45. Funktionen in *MATLAB* können *überladen* werden. Dadurch darf für gleichartige Operationen mit Daten verschiedenen Typs *derselbe Funktionsname* verwendet werden, obwohl es sich nicht um dieselben Funktionen handelt. Ein Programm wird dadurch besser lesbar. Diese Wandlungsfähigkeit wird teilweise in der Literatur als *Polymorphismus* bezeichnet. Polymorphismus ist eine Eigenschaft *objektorientierter* Programmiersprachen, wie z. B. C++. Auch *MATLAB* 6.5 erlaubt die Möglichkeit objektorientierter Programmierung.

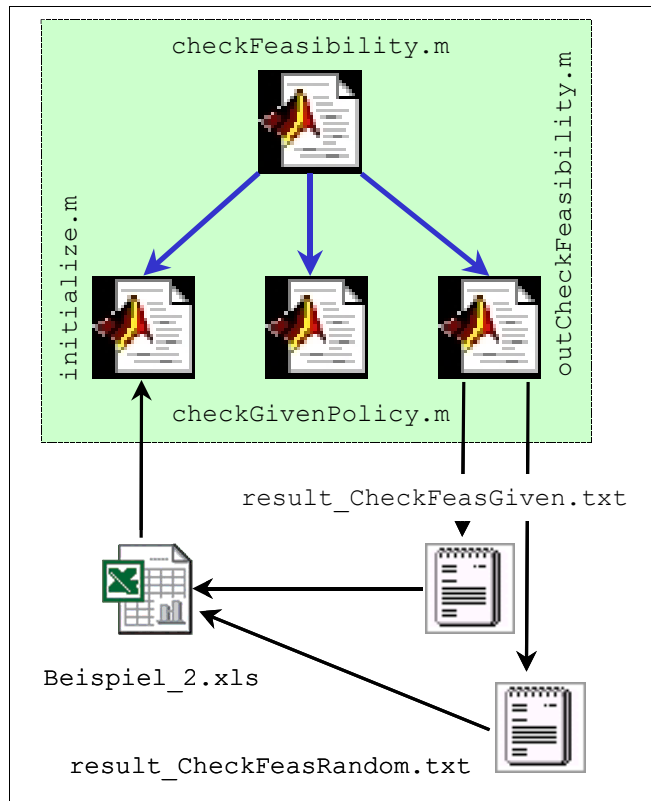


Abbildung 6.16: Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionsaufrufe (blaue Pfeile) im *MATLAB*-Programm (grüner Kasten) bei der Überprüfung einer gegebenen oder mehrerer zufälliger Politiken auf Zulässigkeit für den komplexen dynamischen Akquiseplan.

`outFuzzyDecision` in der Datei `outFuzzyDecision.m` zum Schreiben der Ergebnisse in die Textdatei `result_fuzzyDecision.txt`. Der Satz 6.9 wird von der Funktion

```
function [convergency, normValue, nIter, mueD]
= trafoIter(nX, nT_, nU, PT_, PT, mueC, mueG, tNorm, policy,
omega0, eps, nMax)
```

in der Datei `trafoIter.m` realisiert. In Abb. 6.18 sind die wesentlichen Teile des *MATLAB*-Programmkodes wiedergegeben.

Die **Eingabevariablen** sind Elemente der Liste `parameter`, die von der Funktion `initialize` erzeugt wird (siehe Abschnitt 6.6.3.1):

- `nX`: die Anzahl der Zustände ;
- `nT_`: siehe Seite 326;
- `nU`: siehe Seite 326;
- `PT_`: siehe Seite 324;
- `PT`: siehe Seite 324;
- `mueC`: siehe Seite 324;

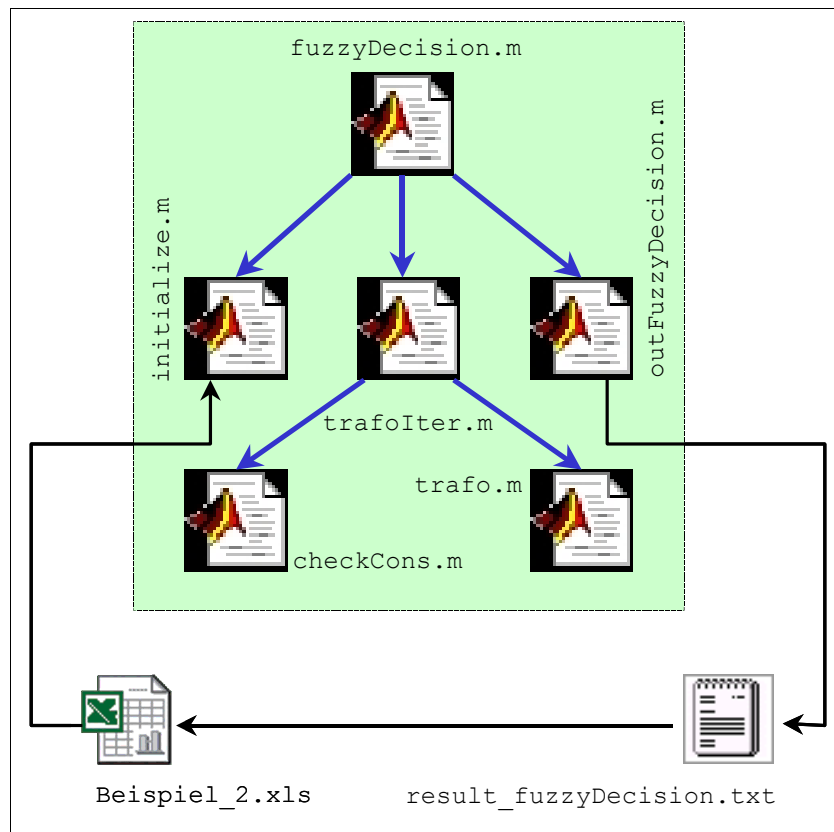


Abbildung 6.17: Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionsaufrufe (blaue Pfeile) im *MATLAB*-Programm (grüner Kasten) bei der Berechnung der unscharfen Entscheidung zu einer gegebenen Politik für den komplexen dynamischen Akquiseplan.

- `mueG`: siehe Seite 324;
- `tNorm`: ein String zur Festlegung der t-Norm mit dem Wertebereich:
 - 'minimum',
 - 'algebraic product',
 - 'bounded difference',
 - 'drastic product',
 - 'Hamacher product',
 - 'Einstein product';
- `policy`: siehe Seite 324;
- `omega0`: siehe Seite 324;
- `eps`: eine nicht negative reellwertige Zahl als Abbruchkriterium (minimale Änderung der Norm) für die Iterationen $n = 1, 2, \dots$ der Transformation (6.96);
- `nMax`: eine positive ganze Zahl als Abbruchkriterium (maximale Anzahl der Iterationen) für die Iterationen $n = 1, 2, \dots$ der Transformation (6.96).

```

% =====%
% Iterate transformation 'trafo' %
% =====%

function [convergency, normValue, nIter, mueD] = trafoIter(nX, nT_, nU,
PT_, PT, mueC, mueG, tNorm, policy, omega0, eps, nMax)

...

% Iterate
omegaOld = omega0;
convergency = 0;
for n = 1:nMax
    omegaNew = trafo(PT_pi, PTpi, mueCpi, mueG, tNorm, omegaOld);
    normValue = norm((omegaNew - omegaOld), inf);
    if (normValue < eps)
        convergency = 1;
        break;
    end
    omegaOld = omegaNew;
end
nIter = n;
mueD = omegaNew;

```

Abbildung 6.18: Auszug aus dem *MATLAB*-Programmcode der Funktion `trafoIter` in der Datei `trafoIter.m`.

Ihre **Rückgabevariablen** sind:

- `convergency`: Kennzeichen („flag“), das angibt, ob die Iteration (6.59) konvergiert (Wert 1) oder nicht (Werte 0);
- `normValue`: Wert der Norm $\|T_{\pi}^n(\omega) - T_{\pi}^{n-1}(\omega)\|_{\infty}$ bei Abbruch der Iteration (entweder durch Konvergenz oder durch Erreichen der maximalen Anzahl der Iterationen);
- `nIter`: Anzahl der Iterationen bei Abbruch der Iteration (entweder durch Konvergenz oder durch Erreichen der maximalen Anzahl der Iterationen);
- `mueD`: der Vektor mit den $|\bar{T}|$ Zugehörigkeitswerten $\mu_D(\pi)$ der durch `policy` gegebenen Politik.

Der **Algorithmus** ist im *MATLAB*-Programmcode in Abb. 6.18 ersichtlich. Wesentlich ist die Schleife `for n = 1:nMax`, in welcher maximal `nMax`-mal die Transformation (6.96) aufgerufen wird, die durch die Funktion

```
function omegaNew = trafo(PT_pi, PTpi, mueCpi, mueG, tNorm,
omega)
```

in der Datei `trafo.m` realisiert ist. In Abhängigkeit des Wertes von `tNorm` wird die Transformation (6.96) mit einer der t -Normen (6.90) – (6.88) berechnet. Die Funktion `checkCons` in der Datei `checkCons.m` trägt nicht zur eigentlichen Berechnung der unscharfen Politik bei, sondern überprüft die Eingabevariablen der Funktion `fuzzyDecision` auf Konsistenz.

6.6.3.4 Verfahren zur Berechnung einer optimalen Politik

Die Grundlage des Verfahrens zur Berechnung einer optimalen Politik eines dynamischen Akquiseplans ist Satz 6.20 im Abschnitt 6.4.3. Dazu wird Alg. 6.6 in die *MATLAB*-Funktion `maxTrafoIter` in der Datei `maxTrafoIter.m` umgesetzt. Eine Übersicht mit dem aufrufenden Programm `optimalPolicy` in der Datei `optimalPolicy.m`, den darin aufgerufenen Funktionen und den Datenflüssen gibt Abb. 6.19. Die Funktion

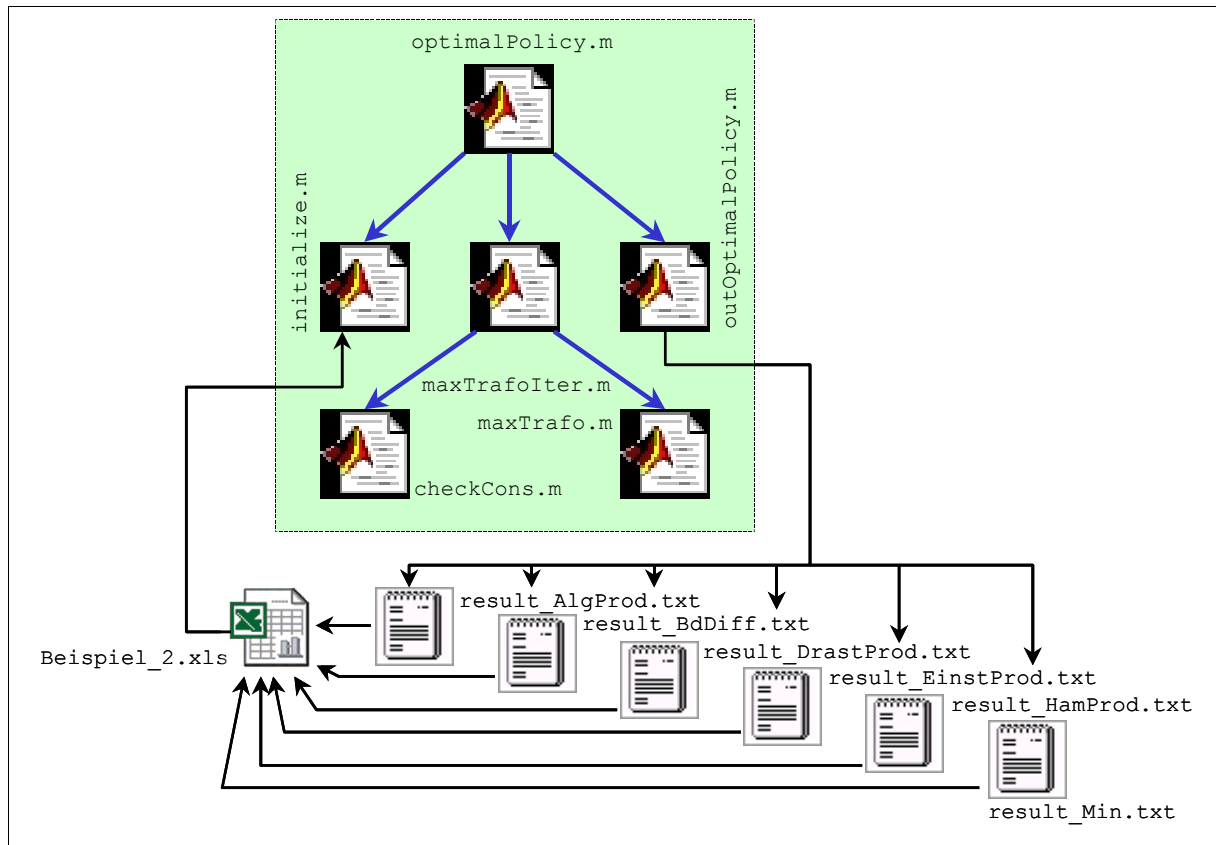


Abbildung 6.19: Schematische Darstellung der Datenflüsse (schwarze Pfeile) und der Funktionsaufrufe (blaue Pfeile) im *MATLAB*-Programm (grüner Kasten) bei der Berechnung einer optimalen Politik für den komplexen dynamischen Akquiseplan.

```
function [convergency, normValue, nIter, mueD, policy, t]
= maxTrafoIter(nX, nT_, nU, PT_, PT, mueC, mueG, tNorm,
omega0, eps, nMax)
```

stimmt in den **Eingabevariablen** und den **Ausgabevariablen** bis auf zwei Unterschiede mit der Funktion `trafoIter` aus Abschnitt 6.6.3.3 überein: Hier ist `policy` keine Eingabevariable, sondern eine Ausgabevariable, und zusätzlich wird mit `t` die Laufzeit der Iteration ausgegeben. In Abb. 6.20 sind die wesentlichen Teile des *MATLAB*-Programmkodes wiedergegeben. In jeder der maximal `nMax` Iterationen gibt die Funktion `maxTrafo` außer dem neuen Wert des Entscheidungsvektors `omegaNew` die Politik zurück, die die Transformation (6.96) maximiert. Bei Konvergenz (Abfrage `if (normValue < eps)`) wird die Iteration abgebrochen (Befehl `break`) und `policy` ist die gesuchte optimale Politik.

```

% =====%
% Iterate transformation 'maxTrafo' %
% =====%

function [convergency, normValue, nIter, mueD, policy, t] = maxTrafoIter(nX, nT_, nU, PT_, PT, mueC, mueG, tNorm, omega0, eps, nMax)

...

% Iterate
tic
omegaOld = omega0;
convergency = 0;
for n = 1:nMax
    [omegaNew, policy] = maxTrafo(nT_, nU, PT_, PT, mueC, mueG, tNorm, omegaOld);
    normValue = norm((omegaNew - omegaOld), inf);
    if (normValue < eps)
        convergency = 1;
        break;
    end
    omegaOld = omegaNew;
end
nIter = n;
mueD = omegaNew;
t = toc

```

Abbildung 6.20: Auszug aus dem *MATLAB*-Programmcode der Funktion `maxTrafoIter` in der Datei `maxTrafoIter.m`.

6.6.4 Bestimmung der Zulässigkeit einer Politik und ihrer unscharfen Entscheidung

Für den einfachen dynamischen Akquiseplan aus Abschnitt 6.6.1 und für den komplexen dynamischen Akquiseplan aus Abschnitt 6.6.2 werden in den folgenden beiden Unterabschnitten jeweils eine gegebene bzw. zufällig erzeugte Politiken auf Zulässigkeit untersucht und für diese Politiken die unscharfen Entscheidungsvektoren berechnet.

6.6.4.1 Einfacher dynamischer Akquiseplan

Für den einfachen dynamischen Akquiseplan in Abschnitt 6.6.1 wird die dort in Nr. 7 in Abb. 6.7 beispielhaft vorgestellte Politik mit dem Programm `checkFeasibility` (vgl. Abschnitt 6.6.3.2) auf Zulässigkeit untersucht. Zugleich wird – Zulässigkeit vorausgesetzt – mit dem Programm `fuzzyDecision` (vgl. Abschnitt 6.6.3.3) für unterschiedliche t -Normen zu dieser Politik die unscharfe Entscheidung berechnet. Die Ergebnisse sind in Abb. 6.21 zusammengestellt.

Interpretation der Ergebnisse in Abb. 6.21

- Links unten ist abzulesen, dass die untersuchte Politik π zulässig ist und bereits die Norm der Übergangsmatrix $P_{\bar{T}}(\pi)$ kleiner als eins ist.

state σ	action α	minimum	algProd	bdDiff	drProd	HamProd	EinProd
1	1	0,048	0,00702	0	0	0,033101	0,003036
2	1	0,18	0,081	0	0	0,131707	0,058484
3	1	0,28	0,126	0	0	0,18806	0,094737
4	1	0,06	0,0324	0	0	0,047788	0,024508
5	2	0,18	0,108	0	0	0,160714	0,081325
6	2	0,36	0,144	0	0	0,233766	0,104046
7	1	0,048	0,0192	0	0	0,040678	0,012565
8	2	0,5	0,27	0,04	0	0,350649	0,219512
9	2	0,27	0,081	0	0	0,165644	0,053607
10	3	0,096	0,01404	0	0	0,066202	0,006073
11	3	0,3	0,135	0	0	0,219512	0,097473
12	3	0,4	0,18	0	0	0,268657	0,135338
13	4	0,3	0,162	0	0	0,238938	0,122542
14	4	0,9	0,9	0,9	0,9	0,9	0,9
15	4	0,9	0,9	0,9	0,9	0,9	0,9
16	3	0,12	0,048	0	0	0,101695	0,031414
17	4	0,9	0,9	0,9	0,9	0,9	0,9
18	4	0,9	0,9	0,9	0,9	0,9	0,9
is feasible:	1	1	1	1	1	1	1
power of PT_:	1	4	4	3	3	4	4
		is convergent					
		# iteration					

Abbildung 6.21: Beispiel_1.xls, Tabelle policy_check. Links oben steht die untersuchte Politik π , gegeben durch die Wertepaare „state/action“, $(\sigma_i, \pi(\sigma_i))$, $i = 1, \dots, 18$.

Links unten steht das Ergebnis der Zulässigkeitsuntersuchung von π . Rechts oben stehen für die verschiedenen t-Normen die unscharfen Entscheidungen μ_D , rechts unten, ob der Algorithmus zur Berechnung der unscharfen Entscheidungen konvergiert und wieviele Iterationen notwendig sind.

- Im rechten Teil stehen oben die unscharfen Entscheidungen zur Politik π und für die in Abschnitt 6.6.3.1 aufgeführten t-Normen: $\mu_D(\pi) = (\mu_D(\sigma_1), \dots, \mu_D(\sigma_{18}))$.
- Satz 6.22 besagt, dass die Iteration konvergiert, wenn π zulässig ist und die t-Norm die Lipschitzbedingung (6.91) erfüllt. Im Abschnitt 6.4.4.1 wird gezeigt, dass mit Ausnahme des drastischen Produkts alle t-Normen die Lipschitzbedingung erfüllen. Rechts unten ist abzulesen, dass das Verfahren zur Berechnung der unscharfen Entscheidungen für alle t-Normen konvergiert, auch für das drastische Produkt. Daran wird anschaulich deutlich, dass die Konvergenzbedingung in Satz 6.22 nur hinreichend, nicht aber notwendig ist.
- Als Abbruchkriterium war für die Berechnung der unscharfen Entscheidungen $\text{eps} = 1\text{E-}12$ vorgegeben (vgl. Abb. 6.18). Trotz des „strengen“ Kriteriums konvergiert das Verfahren für alle t-Normen sehr schnell.
- Die unscharfen Entscheidungen für die t-Normen beschränkte Differenz und drastisches Produkt sind wenig Aussagekräftig, da für die meisten Zustände $x \in \bar{T}$ die Zugehörigkeitswerte $\mu_D(x) = 0$ sind⁴⁶⁾.

46. Dieses Ergebnis bleibt auch bei größerer Darstellungsgenauigkeit erhalten.

-
- Die Werte der unscharfen Entscheidungen hängen von der t-Norm ab.
 - Die Größenverhältnisse zwischen den Werten $\mu_D(\sigma_i)$ und $\mu_D(\sigma_j)$ für $i \neq j$ unterscheiden sich für unterschiedliche t-Normen. Zum Beispiel ist für die Minimumsbildung $\mu_D(\sigma_5) > \mu_D(\sigma_4)$, während für das algebraische Produkt $\mu_D(\sigma_5) < \mu_D(\sigma_4)$ gilt.
 - Den größten Zugehörigkeitswert besitzt die unscharfe Entscheidung μ_D , wenn die Akquise mit einem der Zustände $\sigma_{14}, \sigma_{15}, \sigma_{17}, \sigma_{18}$ beginnt, das heißt wenn der Kunde interessiert ist, sein Alter und sein Bildungsstand bekannt sind und die Entscheidung (Aktion) α_4 getroffen, das heißt ein Angebot mit der Bitte um Bestätigung zugesandt wird (vgl. Seite 320).
 - Dass der Zustand σ_{13} weniger günstig bewertet wird als $\sigma_{14}, \sigma_{15}, \sigma_{17}, \sigma_{18}$, liegt vor allem daran, dass $\mu_G(\sigma_{13}) = 0,6$, während $\mu_G(\sigma_{14}) = \mu_G(\sigma_{15}) = \mu_G(\sigma_{17}) = \mu_G(\sigma_{18}) = 1$ (siehe Abb. 6.6).
 - Dass der Zustand σ_{16} weniger günstig bewertet wird als $\sigma_{14}, \sigma_{15}, \sigma_{17}, \sigma_{18}$, liegt vor allem an der Anwendung der Entscheidung (Aktion) α_3 , das heißt der Zusendung eines Informationsblattes (vgl. Seite 320), das von der unscharfen Nebenbedingung ungünstiger bewertet wird als eine der Aktionen α_1 oder α_2 , verbunden mit den „ungünstigeren“ Übergangswahrscheinlichkeiten $p(x_{t+1} | \sigma_{16}, \alpha_3)$.

6.6.4.2 Komplexer dynamischer Akquiseplan

Wegen der großen Anzahl an Zuständen ist es für den komplexen dynamischen Akquiseplan in Abschnitt 6.6.2 schwierig, eine sinnvolle differenzierte Politik anzugeben. Zur Demonstration des Verfahrens wird eine „triviale“ Politik untersucht – die Politik, die in jedem Zustand ein Angebot zusendet (Entscheidung α_6). Wie im vorigen Abschnitt wird das Programm `checkFeasibility` (vgl. Abschnitt 6.6.3.2) ausgeführt, um die Zulässigkeit der Politik zu ermitteln. Danach wird mit dem Programm `fuzzyDecision` (vgl. Abschnitt 6.6.3.3) für unterschiedliche t-Normen zu dieser Politik die unscharfe Entscheidung berechnet. Die Ergebnisse sind in Abb. 6.22 zusammengestellt.

Interpretation der Ergebnisse in Abb. 6.22

- Links unten ist abzulesen, dass die untersuchte Politik π zulässig ist und bereits die Norm der Übergangsmatrix $P_{\bar{T}}(\pi)$ kleiner als eins ist.
- Im rechten Teil stehen oben die unscharfen Entscheidungen zur Politik π und für die in Abschnitt 6.6.3.1 aufgeführten t-Normen: $\mu_D(\pi) = (\mu_D(\sigma_1), \dots, \mu_D(\sigma_{18}))$.
- Rechts unten ist abzulesen, dass das Verfahren zur Berechnung der unscharfen Entscheidungen für alle t-Normen konvergiert, auch für das drastische Produkt, das die Lipschitzbedingung (6.91).

state σ	action α	minimum	algProd	bdDiff	drProd	HamProd	EinProd	
1	6	0,07714	0,000457	0	0	0,017479	0,000229	
2	6	0,085523	0,000539	0	0	0,019399	0,00027	
3	6	0,072294	0,00041	0	0	0,016347	0,000206	
4	6	0,06	0,000331	0	0	0,014762	0,000166	
5	6	0,06	0,000331	0	0	0,014762	0,000166	
6	6	0,06	0,000331	0	0	0,014762	0,000166	
7	6	0,079917	0,000472	0	0	0,017728	0,000237	
8	6	0,089441	0,000561	0	0	0,019657	0,000281	
9	6	0,074406	0,000419	0	0	0,016581	0,000211	
10	6	0,080531	0,000509	0	0	0,018545	0,000255	
...								
87	6	0,073388	0,001977	0	0	0,024394	0,001043	
88	6	0,057	0,003046	0	0	0,023003	0,001691	
89	6	0,057	0,007616	0	0	0,024708	0,005064	
90	6	0,057	0,00457	0	0	0,02392	0,002684	
91	6	0,085792	0,005001	0	0	0,032882	0,002788	
92	6	0,1025	0,016501	0	0	0,045677	0,011038	
93	6	0,076	0,006093	0	0	0,029798	0,003587	
94	6	0,087115	0,005256	0	0	0,034112	0,002932	
95	6	0,1075	0,01904	0	0	0,051209	0,012758	
96	6	0,076	0,006093	0	0	0,029798	0,003587	
is feasible:	1	1	1	1	1	1	1	is convergent
power of PT_:	1	420	14	3	2	70	11	# iteration

Abbildung 6.22: Beispiel_2.xls, Auszug aus der Tabelle `policy_check` (für die ersten und letzten zehn Zustände). Links oben steht die untersuchte Politik π , gegeben durch die Wertepaare „state/action“, $(\sigma_i, \pi(\sigma_i))$, $i = 1, \dots, 18$. Links unten steht das Ergebnis der Zulässigkeitsuntersuchung von π . Rechts oben stehen für die verschiedenen t-Normen die unscharfen Entscheidungen μ_D , rechts unten, ob der Algorithmus zur Berechnung der unscharfen Entscheidungen konvergiert und wieviele Iterationen notwendig sind.

- Als Abbruchkriterium war für die Berechnung der unscharfen Entscheidungen $\text{eps} = 1\text{E-}12$ vorgegeben (vgl. Abb. 6.18). Für alle t-Normen, mit Ausnahme der Minimum-t-Norm konvergiert das Verfahren sehr schnell.
- Das Verfahren konvergiert für die t-Normen beschränkte Differenz und drastisches Produkt extrem schnell. Die resultierenden unscharfen Entscheidungen sind jedoch unbrauchbar, weil für alle Zustände $x \in \bar{T}$ die Zugehörigkeitswerte $\mu_D(x) = 0$ sind⁴⁷.
- Die Werte der unscharfen Entscheidungen hängen von der t-Norm ab. Dieses Verhalten ist erklärbar, da bei Iteration der Transformation

$$T^t : [0, 1]^k \rightarrow [0, 1]^k$$

$$T^t(\omega) = \bigvee_{\pi \in \Pi} \{ \mu_C(\pi^\nu) \mathbf{t} [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G] \}$$

mit jedem Iterationsschritt gilt

47. Dieses Ergebnis bleibt auch bei größerer Darstellungsgenauigkeit erhalten.

$$\mu_C(\pi^\nu) \mathbf{t} [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G] \leq \min\{\mu_C(\pi^\nu), [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G]\} \quad (6.109)$$

wenn die t-Norm „algebraisches Produkt“ verwendet wird und

$$\mu_C(\pi^\nu) \mathbf{t} [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G] = \min\{\mu_C(\pi^\nu), [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G]\} \quad (6.110)$$

wenn die t-Norm „Minimum“ benutzt wird. In der Regel sind die Übergangswahrscheinlichkeiten der Matrizen $P_{\bar{T}}(\pi^\nu)$ und $P_T(\pi^\nu)$ kleiner als eins. Daher wird in (6.109) in der Regel kein Gleichheitszeichen stehen, sondern die echt-kleiner-Relation. Und deshalb sind die Terme auf der linken Seite in (6.109) bei jedem Iterationsschritt kleiner als in (6.110).

- Die Größenverhältnisse zwischen den Werten $\mu_D(\sigma_i)$ und $\mu_D(\sigma_j)$ für $i \neq j$ unterscheiden sich für unterschiedliche t-Normen. Zum Beispiel sind für die Minimumsbildung die Werte $\mu_D(\sigma_{88})$, $\mu_D(\sigma_{89})$, $\mu_D(\sigma_{90})$ gleich groß, während sie für das algebraische Produkt unterschiedlich groß sind.

6.6.5 Lösung des Optimierungsproblems

Für den einfachen dynamischen Akquiseplan aus Abschnitt 6.6.1 und für den komplexen dynamischen Akquiseplan aus Abschnitt 6.6.2 werden mit dem Algorithmus Alg. 6.4 optimale Politiken berechnet. Der Algorithmus ist im Programm `optimalPolicy` (vgl. Abschnitt 6.6.3.4) implementiert. In den folgenden Unterabschnitten werden verschiedene Aspekte des Verfahrens betrachtet.

6.6.5.1 Laufzeitverhalten

Zur Bewertung des Laufzeitverhaltens wird die Prozessorzeit der Funktion `maxTrafoIter` (siehe Abschnitt 6.6.3.4) auf einem PC mit Betriebssystem Windows XP Professional, Prozessor AMD Athlon XP 1700+ mit 1,47 GHz Taktfrequenz und 1 GB Arbeitsspeicher erfasst, wenn neben dem *MATLAB*-Programm kein anderes Anwendungsprogramm auf dem Prozessor ausgeführt wird. Für den einfachen Akquiseplan sind die Rechenzeiten für die Erfassung zu kurz, das heißt praktisch null. Für den komplexen Akquiseplan ergeben sich die in Abb. 6.23 grafisch dargestellten Laufzeiten. Die Prozessorzeit („CPU time“) ist für alle hier betrachteten t-Normen in Abhängigkeit von der Anzahl der Iterationen bis zum Abbruch des Verfahrens angegeben. Für jede t-Norm wurden zehn Läufe mit unterschiedlichen Anfangsvektoren,

$$\omega^{(0)} = \alpha \cdot \mathbf{1}, \quad (6.111)$$

durchgeführt, wobei $\mathbf{1} \in \mathbb{R}^k$ der Vektor dessen Elemente alle gleich eins sind und $\alpha \in [0, 1]$ (siehe Satz 6.26). Wie zu erwarten war, nimmt die Prozessorzeit mit der Anzahl der Iterationen annähernd linear zu. Auffallend sind die geringen Unterschiede zwischen den t-Normen. In der Mitte des Diagramms ist ersichtlich, dass die t-Normen *Hamacher Produkt* und *Einstein Produkt* etwas langsamer sind als die Minimumbildung. Angesichts mehrerer Multiplikationen und einer Division, die zur Berechnung beider t-Normen nötig sind, überrascht der geringe Unterschied in der Laufzeit. Praktisch ist er unbedeutend.

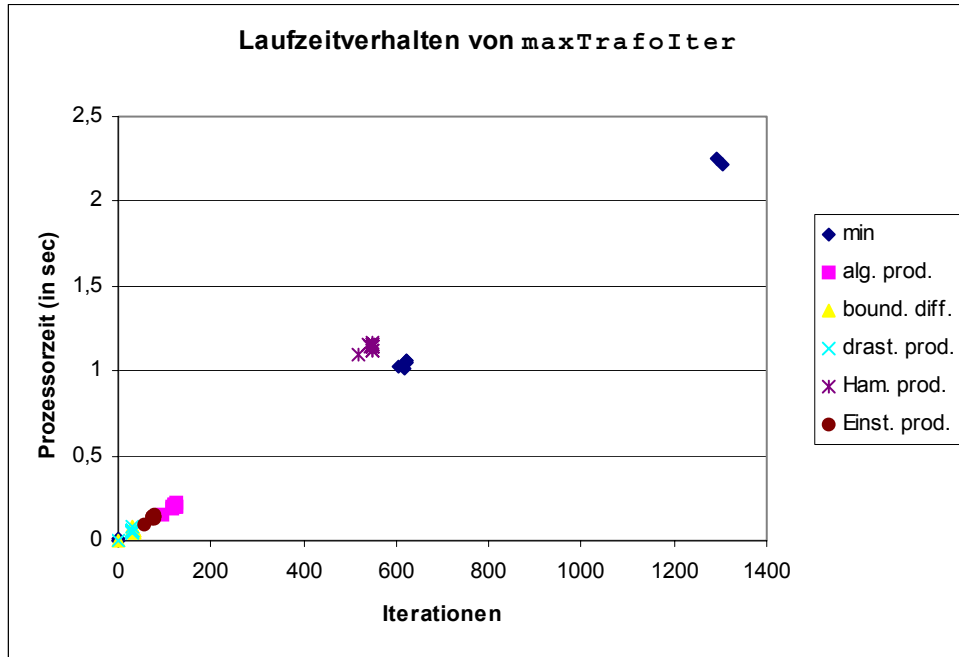


Abbildung 6.23: Laufzeitverhalten der *MATLAB*-Funktion `maxTrafoIter` beim Ausführen des Programms `optimalPolicy` auf dem komplexen dynamischen Akquiseplan, angegeben ist die Prozessorzeit in Abhängigkeit von der Anzahl der Iterationen bis zum Abbruch der Iteration für alle hier untersuchten t-Normen.

6.6.5.2 Konvergenzverhalten

Die Konvergenz des Iterationsverfahrens zur Bestimmung einer optimalen Politik wird mit Hilfe des Weissingerschen Fixpunktsatzes (Satz 6.8) bewiesen. Er besagt, dass das Verfahren mindestens linear konvergiert. Zur Illustration wird das Programm `optimalPolicy` für den komplexen dynamischen Akquiseplan, für alle hier betrachteten t-Normen und für die Startvektoren $\omega^{(0)} = 1 \in \mathbb{R}^k$ und $\omega^{(0)} = 0 \in \mathbb{R}^k$ ausgeführt. Der Wert des Abbruchparameters ist in allen Fällen $\text{eps} = 10\text{E-}6$. Zur Bewertung der Konvergenz werden die folgenden Größen berechnet:

- Der Abstand der unscharfen Entscheidung $\omega^{(n)}$ in der n-ten Iteration vom Fixpunkt $\tilde{\omega}$,

$$\Delta_n := \|\tilde{\omega} - \omega^{(n)}\|_{\infty}, \quad n = 0, 1, \dots, \tilde{n}, \quad (6.112)$$

wobei \tilde{n} die Iteration bezeichnet, bei der das Programm abgebrochen wurde, das heißt $\tilde{\omega} := \omega^{(\tilde{n})}$. Deshalb ist stets $\Delta_{\tilde{n}} = 0$. Die Größe Δ_n wird im Programm als Norm der Differenz, `normDiff` bezeichnet.

- Der Quotient aufeinanderfolgender Werte von `normDiff`,

$$q_{n+1} := \frac{\|\tilde{\omega} - \omega^{(n+1)}\|_{\infty}}{\|\tilde{\omega} - \omega^{(n)}\|_{\infty}}, \quad n = 0, 1, \dots, \tilde{n}, \quad (6.113)$$

die im Programm als Quotient der Normen, normQuot bezeichnet wird.

Für alle t-Normen und für den Startvektor $\omega^{(0)} = 1 \in \mathbb{R}^k$ ist der Verlauf beider Größen in Abb. 6.24 dargestellt. Die Ergebnisse für den Startvektor $\omega^{(0)} = 0 \in \mathbb{R}^k$ zeigt Abb. 6.25. Aus programmtechnischen Gründen ist in den Abbildungen die Iterationsnummer gegenüber obigen Formeln um eins erhöht und dann per Definition $q_1 = 1$ gesetzt.

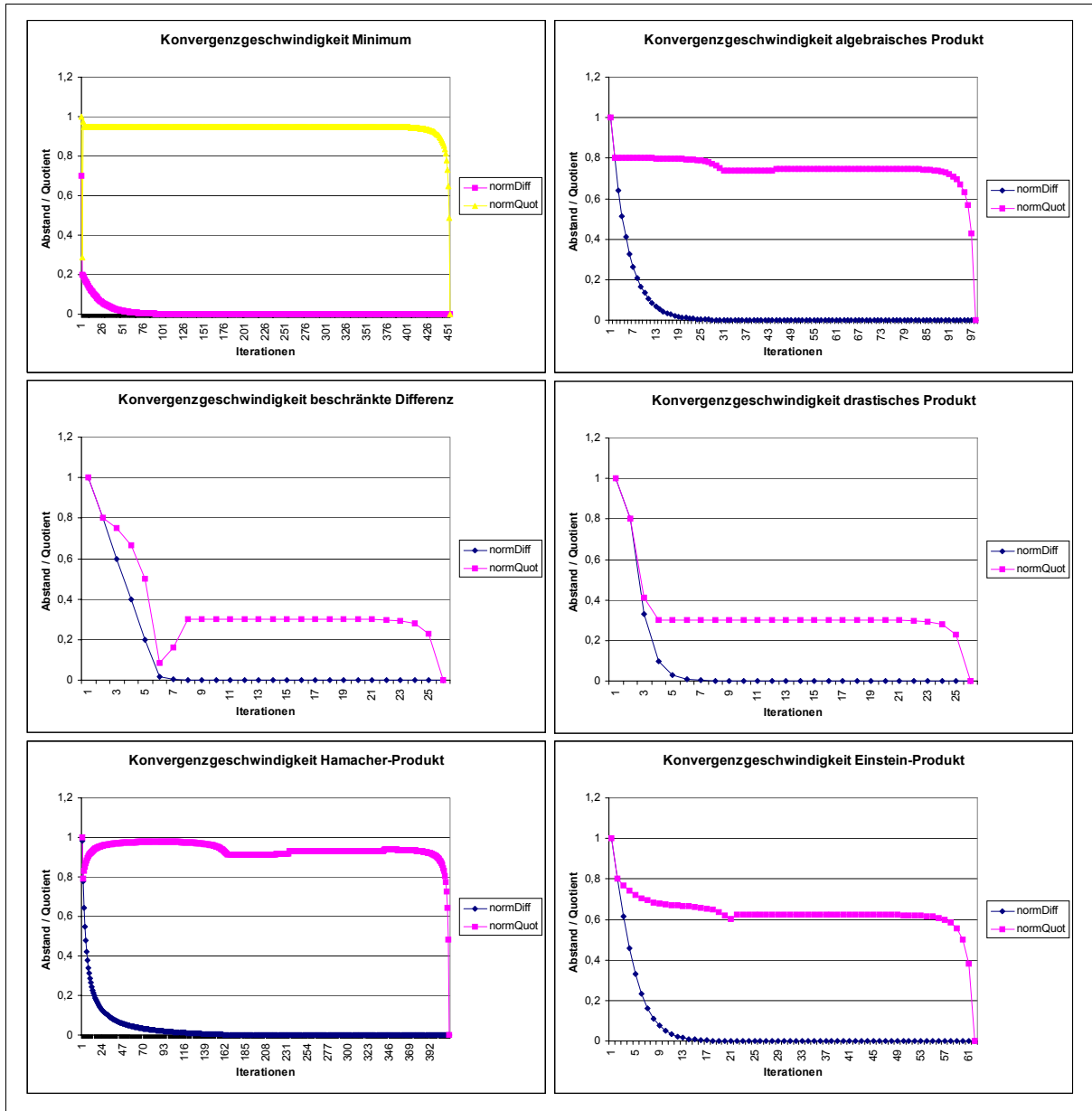


Abbildung 6.24: Darstellung des Verlaufs der Größen normDiff (6.112) und normQuot (6.113) für alle hier betrachteten t-Normen und den Anfangsvektor $\omega^{(0)} = 1 \in \mathbb{R}^k$. Aus programmtechnischen Gründen ist die Nummer der Iteration in den Grafiken gegenüber den Formeln um eins erhöht.

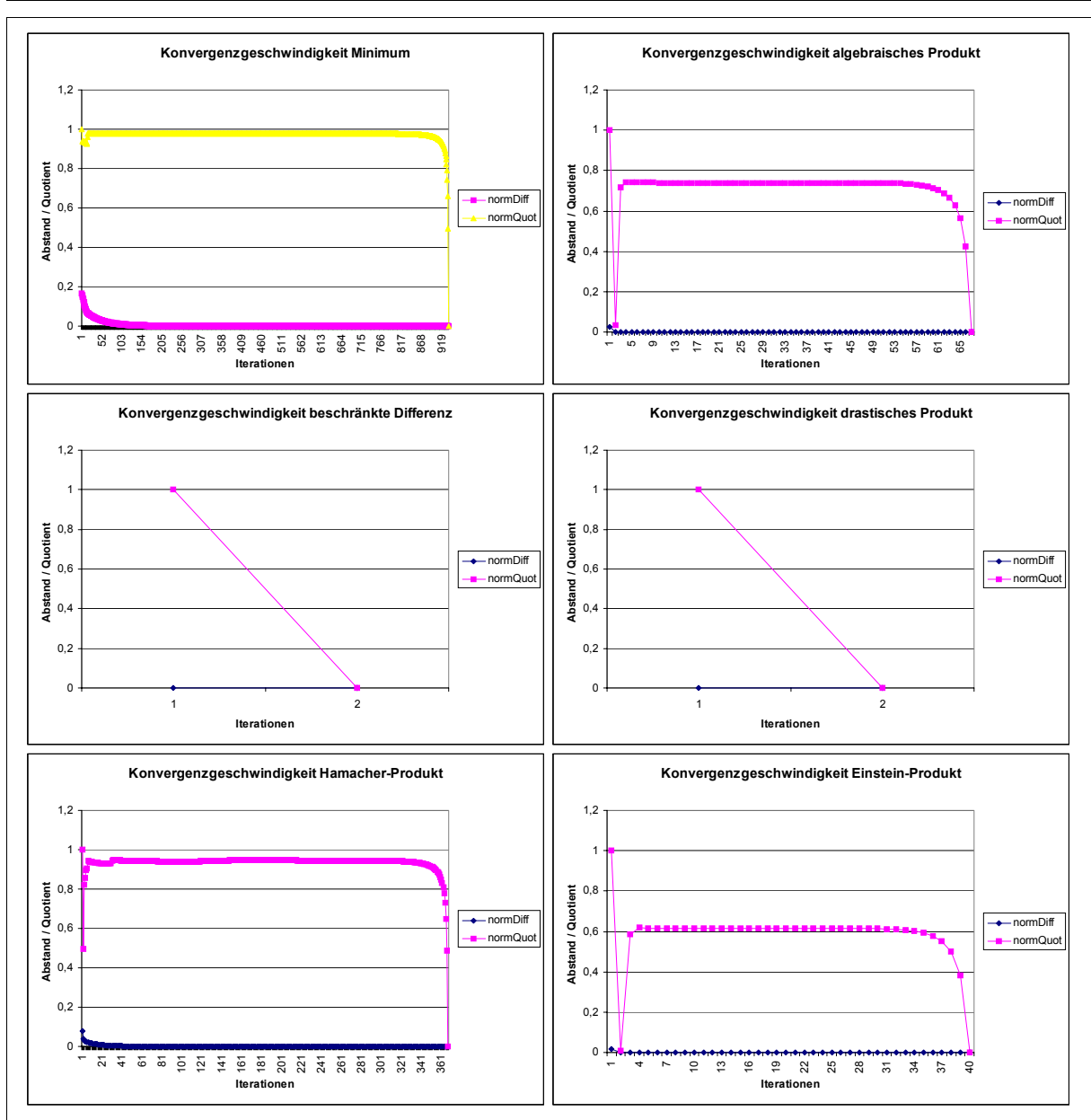


Abbildung 6.25: Darstellung des Verlaufs der Größen normDiff (6.112) und normQuot (6.113) für alle hier betrachteten t-Normen und den Anfangsvektor $\omega^{(0)} = 0 \in \mathbb{R}^k$. Aus programmtechnischen Gründen ist die Nummer der Iteration in den Grafiken gegenüber den Formeln um eins erhöht.

Anmerkungen

1. Die Konvergenz des Verfahrens in allen Fällen wird am monoton fallenden Verlauf von (6.112) deutlich.
2. Der Quotient (6.113) ist in allen Fällen über einen längeren Iterationsbereich ungefähr konstant, was die Linearität der Konvergenz bestätigt.
3. Je größer der Quotient (6.113), das heißt je näher er beim Wert Eins liegt, desto langsamer iteriert das Verfahren, was sich an der Anzahl der Iterationen für den vorgegebenen Wert von ϵ ablesen lässt.

4. Für die t-Normen *beschränkte Differenz* und *drastisches Produkt* konvergiert das Verfahren am schnellsten. Die Konvergenzgeschwindigkeiten der t-Normen *Minimum* und *Hamacher-Produkt* sind etwa gleich groß. Schnell konvergiert das Verfahren für die t-Norm *algebraisches Produkt* und noch schneller für das *Einstein-Produkt*.
5. Die rasche Konvergenz für die t-Normen *beschränkte Differenz* und *drastisches Produkt* wird von den Ergebnissen, der „optimalen“ Politik und der zugehörigen unscharfen Entscheidung relativiert. In allen Fällen ist die „optimale“ Politik für alle Zustände identisch α_1 (Massenbrief) und $\mu_D(x) = 0$ für alle $x \in \bar{T}$.
6. Der dem Verfahren zugrunde liegende Satz 6.26 setzt voraus, dass für die in der Transformation benutzte t-Norm eine Lipschitzbedingung gilt. Wie in Abschnitt 6.4.4.1 gezeigt wird, erfüllt das drastische Produkt die Lipschitzbedingung nicht. Das Verfahren konvergiert dennoch für diese t-Norm. Die Voraussetzung ist also nur hinreichend, nicht aber notwendig.

6.6.5.3 Abhängigkeit der Lösung vom Startwert

Satz 6.26 besagt, dass die Iteration gegen einen eindeutigen unscharfen Entscheidungsvektor $\tilde{\omega}$ konvergiert. Die unscharfe Politik π^{opt} selbst muss nicht eindeutig sein. In der Datei `startwert.xls` sind die Ergebnisse der Ausführung von `optimalPolicy` für alle t-Normen, für die drei Startvektoren $\omega^{(0)}$ gemäß (6.111) mit $\alpha = 0, 1/2, 1$ und für den Abbruchparameter $\text{eps} = 10\text{E-}12$ zusammengestellt. Bezeichne $\tilde{\omega}_\alpha^t$ den Fixpunkt für die t-Norm t und den Startvektor $\omega^{(0)} = \alpha$ und $\pi_{t,\alpha}^{\text{opt}}$ die zugehörige Politik. Dann stellt sich das Ergebnis der Untersuchung wie folgt dar:

- Für $t \in \{\text{alg. Prod.}, \text{beschr. Diff.}, \text{drast. Prod.}, \text{Hamacher Prod.}, \text{Einstein Prod.}\}$ gilt

$$\pi_{t,\alpha}^{\text{opt}} = \pi_{t,\alpha'}^{\text{opt}} \text{ und } \tilde{\omega}_\alpha^t = \tilde{\omega}_{\alpha'}^t \text{ für } \alpha \neq \alpha',$$

das heißt die optimalen Politiken und ihre unscharfen Entscheidungen sind unabhängig vom Startwert. Tatsächlich unterscheiden sich einzelne Komponenten der Vektoren der unscharfen Entscheidungen für unterschiedliche Startwerte in der letzten Nachkommastelle um eine Einheit. Untersuchungen mit unterschiedlichen Werten für eps zeigen, dass es sich dabei um Berechnungsungenauigkeiten oder Anzeigungsungenauigkeiten handelt.

- Für $t = \text{Minimum}$ gelten die Aussagen der anderen t-Normen nicht, das heißt

$$\pi_{t,0}^{\text{opt}} \neq \pi_{t,0,5}^{\text{opt}}, \pi_{t,0,5}^{\text{opt}} \neq \pi_{t,1}^{\text{opt}} \text{ und } \pi_{t,0}^{\text{opt}} \neq \pi_{t,1}^{\text{opt}}$$

sowie

$$\tilde{\omega}_0^t \neq \tilde{\omega}_{0,5}^t, \tilde{\omega}_{0,5}^t \neq \tilde{\omega}_1^t \text{ und } \tilde{\omega}_0^t \neq \tilde{\omega}_1^t.$$

Die Ergebnisse für die t-Norm *Minimum* widersprechen nur auf den ersten Blick der Aussage des Satzes Satz 6.26. Eine Überprüfung der Politiken $\pi_{t,0}^{\text{opt}}, \pi_{t,0,5}^{\text{opt}}, \pi_{t,1}^{\text{opt}}$ mit dem Programm

checkFeasibility auf Zulässigkeit ergibt nämlich, dass diese Politiken *nicht* zulässig sind. Damit gibt es in der Menge der Politiken Π des untersuchten komplexen dynamischen Akquiseplans auch nicht zulässige Politiken, und deshalb gilt die Voraussetzung des Satzes, $\Pi = \Pi_p$ nicht. Dass für die anderen t-Normen dennoch durch Iteration eine optimale Politik gefunden wird, die nicht vom Startwert abhängt, zeigt ein weiteres Mal, dass die Voraussetzungen des Satzes nur hinreichend sind und nicht notwendig.

Anmerkung

Durch eine Erweiterung des Zustandsraums kann gewährleistet werden, dass $\Pi = \Pi_p$ zutrifft (siehe Abschnitt 6.5.3).

6.6.5.4 Abhängigkeit der Lösung von der verwendeten t-Norm

Untersuchungen zeigen, dass die optimale Politik von der t-Norm abhängt. Das zeigen die in der Datei `startwert.xls` zusammengestellten Ergebnisse in Abschnitt 6.6.5.3 für den komplexen dynamischen Akquiseplan und die Ergebnisse des Programm `optimalPolicy` für den einfachen dynamischen Akquiseplan in Abb. 6.26. Die Ursache dafür ist die Definition

state σ	Policy π					
	minimum action α	alg. prod. action α	bounded diff. action α	drastic prod. action α	Hamacher action α	Einstein action α
1	2	1	1	1	1	1
2	2	1	1	1	1	1
3	1	1	1	1	1	1
4	2	2	1	1	2	2
5	1	1	1	1	1	1
6	2	1	1	1	1	1
7	2	1	1	1	2	1
8	2	2	1	1	2	2
9	1	1	1	1	1	1
10	2	2	2	1	2	2
11	2	2	2	1	2	2
12	2	2	2	1	2	2
13	2	2	2	1	2	2
14	4	4	4	4	4	4
15	4	4	4	4	4	4
16	2	2	2	1	2	2
17	4	4	4	4	4	4
18	4	4	4	4	4	4
stop criterion	1E-12	1E-12	1E-12	1E-12	1E-12	1E-12
stop value	0	0	0	0	0	0
# iterations	3	4	4	4	4	4
CPU time (sec)	0	0	0	0	0	0

Abbildung 6.26: Auszug aus `Beispiel_1.xls`, Tabelle `policy_result`. Optimale Politiken (gegeben durch die Wertepaare (σ, α)) für alle t-Normen, den Startvektor $\omega^{(0)} = 1$ und `eps = 1E-12` (*stop criterion*). Alle diese Politiken sind auch zulässig.

der Halbordnung der Politiken über ihre *unscharfen Entscheidungen* (siehe Def. 6.13). Letztere hängen von der t-Norm ab (siehe (6.97)):

$$T^t : [0, 1]^k \rightarrow [0, 1]^k$$

$$T^t(\omega) = \bigvee_{\pi \in \Pi} \{ \mu_C(\pi^\nu) \mathbf{t} [P_{\bar{T}}(\pi^\nu) \cdot \omega + P_T(\pi^\nu) \cdot \mu_G] \} \quad (6.114)$$

Im Beispiel stimmen die optimalen Politiken für die t-Normen algebraisches Produkt und Einstein-Produkt überein. Zwischen der optimalen Politik für das algebraische Produkt und das Hamcher-Produkt gibt es nur einen geringen Unterschied. Die Politiken für das Minimum, die beschränkte Differenz und das drastische Produkt weichen von den anderen Politiken stark ab.

Zur Verdeutlichung der Abhängigkeit von der t-Norm sind in Abb. 6.27 die unscharfen Entscheidungen $\omega^{(n)}$ und die zugehörigen maximierenden Politik $\pi^{(n)}$ für die Iterationen der Transformation (6.114) und die t-Normen *algebraisches Produkt* und *Hamacher-Produkt* dargestellt. Die optimalen Politiken beider t-Normen unterscheiden sich nur für den Zustand σ_7 . Für das algebraische Produkt ist $\pi(\sigma_7) = \alpha_1$, für das Hamacher-Produkt ist $\pi(\sigma_7) = \alpha_2$. Dieser Unterschied entsteht in der dritten Iteration. Während für das Hamacher-Produkt der Wert von $\omega^{(n)}(\sigma_7)$ für $n = 2, 3, 4$ gleich bleibt, wird er für das algebraische Produkt in der dritten Iteration durch den Wechsel von $\pi(\sigma_7) = \alpha_2$ zu $\pi(\sigma_7) = \alpha_1$ noch einmal größer, $\omega^{(3)}(\sigma_7) > \omega^{(2)}(\sigma_7)$.

6.6.5.5 Fachliche Interpretation der optimalen Politiken

Für das im vorigen Abschnitt verwendete Beispiel sind in Abb. 6.28 für alle t-Normen die optimalen Politiken und ihre unscharfen Entscheidungen dargestellt. Wie bereits festgestellt, hängt die optimale Politik von der t-Norm ab. In der Abbildung ist zu sehen, dass für die t-Norm *drastisches Produkt* für fast alle Zustände die unscharfe Entscheidung gleich null ist. Die Kunden, die zu einem dieser Zustände gehören werden im Mittel keinen „Nutzen“ erzeugen. Diese Bewertung widerspricht der „praktischen Erwartungshaltung“, weil alle Politiken des Beispiels zulässig sind, somit die Endzustände mit Wahrscheinlichkeit eins erreicht werden können und die Hälfte der Endzustände durch das unscharfe Ziel μ_B positiv bewertet wird. Somit steht in Frage, ob das *drastische Produkt* eine für praktische Zwecke geeignete t-Norm ist. Neben der – wie zuvor schon festgestellten – fehlenden Lipschitzbedingung (6.91) spricht vor allem die Definition (6.89) dieser t-Norm gegen ihre praktische Verwendung. So widerspricht sie der Idee von Zadeh und Bellman, eine unscharfe Entscheidung als „Zusammenkommen“ von unscharfem Ziel und unscharfer Nebenbedingung zu definieren (vgl. Def. 6.6), denn diese t-Norm erlaubt „Abstriche“ an den Nebenbedingungen nur, wenn das unscharfe Ziel eins ist, und umgekehrt sind „Abstriche“ am unscharfen Ziel nur zulässig, wenn die Nebenbedingungen gänzlich eingehalten werden. In allen anderen Fällen wird die unscharfe Entscheidung gleich null. Aus Anwendungssicht ist es aber gewollt, wenn eine unscharfe Nebenbedingung mit $\mu_C = 0,9$ und ein unscharfes Ziel mit $\mu_G = 0,7$ nicht gänzlich abgelehnt werden, wie dies durch eine unscharfe Entscheidung $\mu_D = 0$ indiziert wird. In Beispielen mit dem komplexen dynamischen Akquiseplan zeigt die beschränkte Differenz ein ähnliches Verhalten.

	algebraisches Produkt						Hamacher-Produkt					
	Zustand	Iterationen					Zustand	Iterationen				
unscharfe Entscheidung	1	0	0,005	0,225	0,315	0,315	1	0	0,034483	0,31034	0,32474	0,32474
	2	0	0,007	0,27	0,378	0,378	2	0	0,042945	0,34615	0,38969	0,38969
	3	0	0,007	0,18	0,441	0,441	3	0	0,042945	0,26866	0,45464	0,45464
	4	0	0,006	0,18	0,18	0,18	4	0	0,038961	0,26866	0,26866	0,26866
	5	0	0,01	0,126	0,126	0,126	5	0	0,052632	0,16711	0,16711	0,16711
	6	0	0,01	0,189	0,189	0,189	6	0	0,052632	0,242	0,242	0,242
	7	0	0,006	0,216	0,252	0,252	7	0	0,038961	0,26866	0,26866	0,26866
	8	0	0,01	0,27	0,27	0,27	8	0	0,052632	0,35065	0,35065	0,35065
	9	0	0,01	0,378	0,378	0,378	9	0	0,052632	0,43852	0,43852	0,43852
	10	0	0,315	0,63	0,63	0,63	10	0	0,37725	0,64948	0,64948	0,64948
	11	0	0,252	0,63	0,63	0,63	11	0	0,32391	0,64948	0,64948	0,64948
	12	0	0,189	0,63	0,63	0,63	12	0	0,25506	0,64948	0,64948	0,64948
	13	0	0,162	0,63	0,63	0,63	13	0	0,23894	0,64948	0,64948	0,64948
	14	0	0,9	0,9	0,9	0,9	14	0	0,9	0,9	0,9	0,9
	15	0	0,9	0,9	0,9	0,9	15	0	0,9	0,9	0,9	0,9
	16	0	0,162	0,63	0,63	0,63	16	0	0,23894	0,64948	0,64948	0,64948
	17	0	0,9	0,9	0,9	0,9	17	0	0,9	0,9	0,9	0,9
	18	0	0,9	0,9	0,9	0,9	18	0	0,9	0,9	0,9	0,9
Politik	1		4	2	1	1	1		4	2	1	1
	2		4	2	1	1	2		4	2	1	1
	3		4	2	1	1	3		4	2	1	1
	4		4	2	2	2	4		4	2	2	2
	5		4	1	1	1	5		4	1	1	1
	6		4	1	1	1	6		4	1	1	1
	7		4	2	1	1	7		4	2	2	2
	8		4	2	2	2	8		4	2	2	2
	9		4	1	1	1	9		4	1	1	1
	10		4	2	2	2	10		4	2	2	2
	11		4	2	2	2	11		4	2	2	2
	12		4	2	2	2	12		4	2	2	2
	13		4	2	2	2	13		4	2	2	2
	14		4	4	4	4	14		4	4	4	4
	15		4	4	4	4	15		4	4	4	4
	16		4	2	2	2	16		4	2	2	2
	17		4	4	4	4	17		4	4	4	4
	18		4	4	4	4	18		4	4	4	4

Abbildung 6.27: Übersicht der unscharfen Entscheidungen $\omega^{(n)}$ (oben) und der zugehörigen maximierenden Politik $\pi^{(n)}$ (unten) für die Iterationen der Transformation (6.114) und die t-Normen *algebraisches Produkt* und *Hamacher-Produkt*. In dem farblich unterlegten Bereich kann nachvollzogen werden, wie die unterschiedlichen optimalen Politiken entstehen.

Für alle anderen t-Normen stimmen im Beispiel die Politiken für nicht-beendete Zustände mit Produktinteresse „interessiert“ überein. Bei den anderen nicht-beendeten Zuständen, das sind jene mit Produktinteresse „unbekannt“, stimmen die Ergebnisse für die t-Normen *algebraisches Produkt*, *Hamacher-Produkt* und *Einstein-Produkt* weitgehend überein, was an ihrer „Ähnlichkeit“ liegt, die in den Funktionsgraphen augenscheinlich wird (siehe Abb. 6.29). Nur die t-Norm *Minimum* weicht deutlich davon ab. Nachfolgend werden die Ergebnisse für das Minimum und das algebraische Produkt näher betrachtet.

State	Age	Education	Interest	minimum		alg. Product		bounded diff.		drastic product		Hamacher prod.		Einstein prod.	
				π_{opt}	μ_{opt}	π_{opt}	μ_{opt}	π_{opt}	μ_{opt}	π_{opt}	μ_{opt}	π_{opt}	μ_{opt}	π_{opt}	μ_{opt}
1	unknown	unknown	unknown	2	0,45	1	0,315	1	0,3	1	0	1	0,32474	1	0,30583
2	unknown	low	unknown	2	0,45	1	0,378	1	0,36	1	0	1	0,38969	1	0,36699
3	unknown	high	unknown	1	0,49	1	0,441	1	0,42	1	0	1	0,45464	1	0,42816
4	young	unknown	unknown	2	0,4	2	0,18	1	0,12	1	0	2	0,26866	2	0,13534
5	young	low	unknown	1	0,18	1	0,126	1	0	1	0	1	0,16711	1	0,10112
6	young	high	unknown	2	0,36	1	0,189	1	0	1	0	1	0,242	1	0,15505
7	old	unknown	unknown	2	0,3	1	0,252	1	0,24	1	0	2	0,26866	1	0,24466
8	old	low	unknown	2	0,5	2	0,27	1	0,06	1	0	2	0,35065	2	0,21951
9	old	high	unknown	1	0,54	1	0,378	1	0,24	1	0	1	0,43852	1	0,33216
10	unknown	unknown	interested	2	0,7	2	0,63	2	0,6	1	0	2	0,64948	2	0,61165
11	unknown	low	interested	2	0,7	2	0,63	2	0,6	1	0	2	0,64948	2	0,61165
12	unknown	high	interested	2	0,7	2	0,63	2	0,6	1	0	2	0,64948	2	0,61165
13	young	unknown	interested	2	0,7	2	0,63	2	0,6	1	0	2	0,64948	2	0,61165
14	young	low	interested	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9
15	young	high	interested	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9
16	old	unknown	interested	2	0,7	2	0,63	2	0,6	1	0	2	0,64948	2	0,61165
17	old	low	interested	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9
18	old	high	interested	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9	4	0,9

Abbildung 6.28: Auszug aus Beispiel_1.xls, Tabelle result. Optimale Politiken und ihre unscharfen Entscheidungen (gegeben durch die Wertepaare (σ, α)) für alle t-Normen, den Startvektor $\omega^{(n)} = 0$ und $\epsilon_{ps} = 1E-12$ (stop criterion). (Das Beispiel stimmt mit dem in Abb. 6.26 überein.)

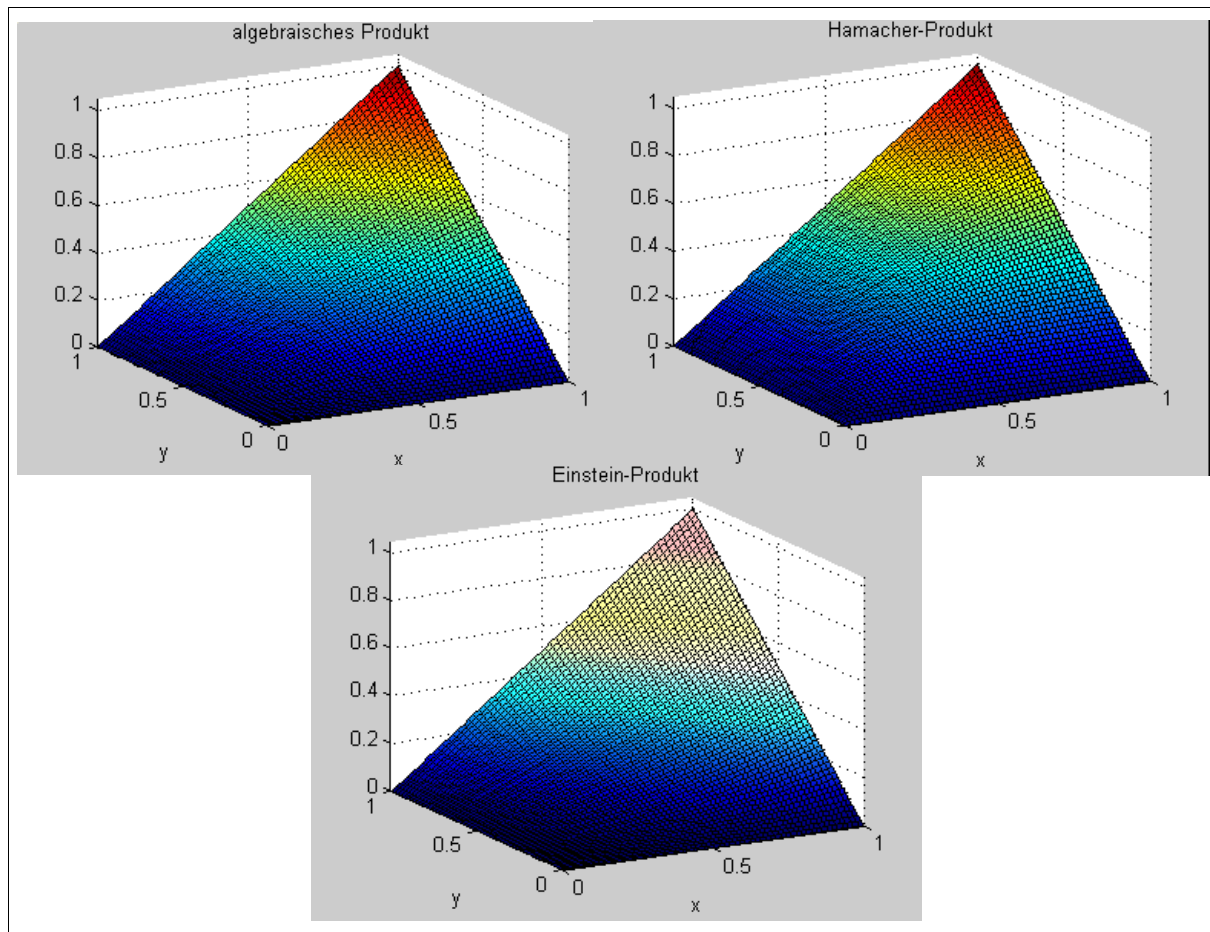


Abbildung 6.29: Funktionsgraphen der t-Normen *algebraisches Produkt*, *Hamacher-Produkt* und *Einstein-Produkt*.

Minimum und algebraisches Produkt

Zunächst fällt auf, dass beide optimale Politiken für keinen der nicht-beendeten Zustände die Aktion α_3 wählt:

- α_3 : Brief mit einem *Informationsblatt* und Antwortkupon mit zwei Reaktionsmöglichkeiten für den Kunden: entweder Anforderung eines konkreten Angebots (was die Angabe des Alters erfordert) oder Bitte um einen Hausbesuch (ohne nähere Angaben).

Die Ursache liegt in einer Eigenschaft der Entscheidung α_3 , die diese von allen anderen Entscheidungen unterscheidet: In jedem Zustand gibt es eine Aktion $u \in U \setminus \{\alpha_3\}$ für die gilt:

$$\mu_C(u | x) > \mu_C(\alpha_3 | x) \text{ für alle } x \in X.$$

Diese Eigenschaft besitzt keine der anderen Entscheidungen (siehe Abb. 6.5). Ansonsten zeigen beide Politiken in grober Näherung folgende Gemeinsamkeiten:

1. Wenn das Produktinteresse „interessiert“ ist und die Werte der beiden anderen Attribute bekannt ist, dann wird α_4 gewählt.

Diese Wahl entspricht der Einschätzung, dass das Zusenden eines Angebots erst lohnt, wenn ein potenzieller Kunde interessiert und über ihn möglichst viel bekannt ist. Zu dieser Wahl trägt auch bei, dass diese Zustände von der unscharfen Nebenbedingung mit dem Zugehörigkeitswert eins bewertet werden (siehe Abb. 6.5).

2. Wenn das Produktinteresse „interessiert“ ist und für eines der beiden anderen Attribute der Wert „unbekannt“ ist, dann wird α_2 gewählt.

Diese Wahl entspricht der Einschätzung, dass ein interessierter Kunde auf einen Telefonanruf nicht ungehalten reagieren wird, und dass er in der Regel am besten durch ein persönliches Gespräch überzeugt werden kann. In diesen Zuständen sind die Werte der unscharfen Nebenbedingungen besser als für die anderen Aktionen.

3. Wenn das Produktinteresse „unbekannt“ ist, dann wird mit t-Norm algebraisches Produkt in der Regel α_1 gewählt. Mit der t-Norm Minimum werden gleichermaßen α_1 und α_2 gewählt.

Diese Wahl entspricht die Einschätzung, dass ein Kunde, der möglicherweise nicht interessiert ist, keinen Telefonanruf wünscht. Dann ist der Massenbrief ein gutes Medium zur aufwandsarmen Kontaktaufnahme. Er wird in den Nebenbedingungen beim Minimum höher bewertet als alle anderen Entscheidungen. Dass die Entscheidung nicht alleine an den Nebenbedingungen hängt, zeigt die teilweise Auswahl von α_2 .

Zusammenfassend lässt sich feststellen, dass die optimalen Politiken in grober Näherung verständlich sind, auch wenn nicht alle Einzelentscheidungen im Detail verstanden werden können.

7 Überblick, Bewertung und Ausblick

In diesem abschließenden Kapitel werden der Gegenstand der vorliegenden Arbeit und seine Teilaspekte zusammengefasst und bewertet. Davon ausgehend werden mögliche Erweiterungen der Arbeit beschrieben sowie mögliche Anwendungsgebiete. Anstelle eines streng chronologischen Vorgehens mit Verweisen auf die einzelnen Abschnitte der Arbeit erfolgt die Zusammenfassung thematisch und verweist auf Definitionen, Sätze, Abbildungen, Tabellen und auch Abschnitte, je nachdem welche Darstellung das jeweilige Thema am besten illustriert.

7.1 Rückblick und Bewertung

Der Gegenstand dieser Arbeit – *Unscharfe stochastische Optimierung und Anwendungen im Marketing* – ist das Ergebnis der Zusammenarbeit des Lehrstuhls für Numerische und Angewandte Mathematik an der Brandenburgischen Technischen Universität Cottbus und des Software-Hauses *Format* in Frankfurt an der Oder, das sich auf Entwicklung und Vertrieb von Software für Finanzdienstleister, kleine bis mittelgroße Banken und Versicherungsagenturen spezialisiert hat. Im Mittelpunkt des Interesses stand dabei das Software-Werkzeug *AkquiSys*, das die Planung, Organisation, Durchführung und Auswertung von Kunden-Akquisitionen unterstützt, die in Form einer Kampagne mit festgelegten Akquiseaktionen durchgeführt werden. Dieses System galt es zu einem Entscheidungsunterstützungssystem („decision support system“) zu erweitern, wodurch gleichsam generisch die in dieser Arbeit behandelten Themen *Simulation* und *Optimierung* in den Mittelpunkt der Betrachtungen gerückt sind. Bevor nachfolgend die Teilaspekte der unscharfen stochastischen Optimierung und ihrer Anwendungen im Marketing unter Nennung ihrer wichtigsten Ergebnisse im Überblick aufgeführt werden, seien drei Aspekte betont:

1. Die Arbeit ist praxisorientiert. Das ist aus oben Gesagtem ersichtlich.
2. Alle in dieser Arbeit entwickelten Verfahren sind mit Standardsoftware implementiert und mit wirklichkeitsnahen Beispielen getestet worden¹⁾. Dazu gehören:
 - Simulation des Verlaufs der Kundenakquise mit einem einfachen Akquiseplan und der damit einhergehenden Entwicklung einiger relevanter ökonomischer Kenngrößen; der Simulations-Algorithmus Alg. 3.1 wurde in *Maple* programmiert (Abb. 3.9);
 - Auslastungsoptimale Beschickung eines Akquiseplans; die Algorithmen sind Bestandteile eines umfangreichen über Parameter steuerbaren C++-Programms (Abb. 4.15);
 - Berechnung einer optimalen Politik für die Kundenakquise mit einem dynamischen Akquiseplan; der Algorithmus (Alg. 6.4) wurde in *Matlab* programmiert (Abb. 6.20).
3. Die Verfahren und ihre Programme können zu einem einheitlichen Softwaresystem für die Entscheidungsunterstützung bei der Kundenakquise weiter entwickelt werden.

Auf den letzten Punkt wird im Abschnitt 7.2.4 eingegangen. Nachfolgend werden die Teilaspekte der Arbeit aufgelistet und ihre wichtigsten Ergebnisse beschrieben.

1. Anmerkungen zu den Programmiersprachen finden sich in den Abschnitten, wo die Programme beschrieben werden.

Unschärfe Logik

Unschärfe Logik oder genauer die *Theorie unscharfer Mengen* stellt die formalen Mittel für die Darstellung von *Unschärfe* bereit. Ihre Einführung im Kapitel 2 umfasst in den Definitionen 2.2 – 2.15 die zum Verständnis nötigen grundlegenden Begriffe, deren Darstellung sich an der „strengen Form“ von [66] orientiert. Der darüber hinaus gehende neue Beitrag dieser Arbeit zur unscharfen Logik besteht in zweierlei:

1. Aus dem Anwendungsbezug ergibt sich die Notwendigkeit, im Kapitel 4 einige neue Begriffe einzuführen. Hierzu zählen:
 - die unscharfe Kapazitätsschranke (Def. 4.2),
 - die verallgemeinerte Umkehrabbildung einer Zugehörigkeitsfunktion (Def. 4.3) und
 - das scharfe Äquivalent eines unscharfen Optimierungsproblems (Def. 4.4).
2. In den Abschnitten 2.2 – 2.4 geht die Arbeit ausführlich auf die Auseinandersetzung um die Existenzberechtigung der unscharfen Logik ein. Da auch im 40sten Jahr ihres Bestehens (2005) diese Diskussion nicht beendet ist, hielt der Verfasser einige Anmerkungen dazu für gerechtfertigt. Interessanterweise findet die unscharfe Logik „prominente Unterstützer“ schon lange bevor sie von Zadeh erdacht wurde, z. B. auf dem Gebiet der Philosophie und in der Mathematik: Engels’ *Anti-Dühring* bezieht sich derart konkret auf das Problem der Unschärfe, dass er vermutlich der unscharfen Logik einen Abschnitt in seinem Buch gewidmet hätte, wäre sie damals bekannt gewesen. Ein anderer Unterstützer ist der Mathematiker Lukasiewicz. Die Ähnlichkeit seiner Ideen mit denen von Zadeh [177] und die Parallelen der um seine mehrwertige Logik geführten Auseinandersetzung mit der Auseinandersetzung um die unscharfe Logik, auch in Abgrenzung zur Wahrscheinlichkeitstheorie, wurde bislang in der Literatur wenig beachtet und ist in der vorliegenden Arbeit herausgearbeitet worden.

Unschärfe Mengen werden in diese Arbeit eingeführt, weil die zu modellierenden Probleme mit Begriffen operieren, die in der Wirklichkeit nicht scharf sind, z. B. die Anzahl von Angeboten, die ein Versicherungsagent oder -sachbearbeiter an einem Arbeitstag schreiben kann, oder die Eignung einer Akquiseaktion für einen bestimmten Kunden.

Simulation der Kundenakquisition mit Akquiseplänen

Das eingangs erwähnte Software-Werkzeug *AkquiSys* (Abschnitt 3.1.1) liefert auch die Grundlage des Modells der Kundenakquisition: den *Akquiseplan*. Er wird zunächst in Def. 3.1 als gerichteter Graph mit bestimmten Eigenschaften formalisiert. Sein Einsatz ist in *AkquiSys* auf die Festlegung von Akquiseaktionen und die Beobachtung des Kundenverhaltens beschränkt. Um die Simulation des Kundenverhaltens zu ermöglichen, wird in den Abschnitten 3.2.1 – 3.2.2 ein Wahrscheinlichkeitsraum eingeführt. Mit ihm kann dem Ereignis „*ein Kunde befindet sich zum Zeitpunkt t im Knoten A_i* “ (Def. 3.5), das gleichbedeutend ist mit: „*auf einen Kunden wird zum Zeitpunkt t die Akquiseaktion A_i angewandt*“, eine Wahrscheinlichkeit zugewiesen werden, die so genannte *Präsenzwahrscheinlichkeit* (Def. 3.6). Damit dieser Wahrscheinlichkeitsraum „praktisch nutzbar“ ist, werden drei wesentliche Eigenschaften gezeigt:

1. Der Wahrscheinlichkeitsraum ist durch die Angabe zweier Arten von Größen vollständig festgelegt (Satz 3.6):

- der zeitlichen Punktwahrscheinlichkeiten des Knotenwechsels (Def. 3.9) und
- der Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten (Def. 3.10).

Mit der Erfahrung eines Fachmanns können ein Akquiseplan und diese Größen mit geringem Aufwand festgelegt werden. Ein solcherart entstandener Akquiseplan (Abb. 3.5) wird in vorliegender Arbeit mehrfach verwendet, ebenso die in den Definitionsdateien `*.tme` und `*.tms` (siehe Abschnitt 4.4.1.2) festgelegten realistischen zeitlichen Punktwahrscheinlichkeiten und Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten.

2. Aus obigen Größen lassen sich iterativ in der Zeit fortschreitend die
 - Präsenzwahrscheinlichkeiten (siehe oben) und
 - die zeitabhängigen Übergangsmatrizen
 berechnen (Satz 3.12). Den zugehörigen Algorithmus zeigt Alg. 3.1.
3. Wenn die oben genannten zeitlichen Punktwahrscheinlichkeiten einen endlichen Träger besitzen, dann ist die Akquise von *endlicher* Dauer (Satz 3.16).

Dank diesen Eigenschaften kann das zeitliche Kundenverhalten, der „Kundenfluss“, durch den Akquiseplan während einer Akquise simuliert werden. Der Nutzen der Simulation wird durch Einführung einer Ertragsfunktion (Def. 3.14) erweitert. Mit ihrer Hilfe wird die zeitliche Entwicklung der Kosten und Erträge simuliert. Der zugehörige Algorithmus wurde mit *Maple*²⁾ programmiert. Im Abschnitt 3.3 werden in vier Fallbeispielen unterschiedliche Kosten-/Ertragsszenarien simuliert und interpretiert. Mit Hilfe der Simulation können aber auch Fragen wie:

- „Wann erreicht die Akquise den Kostendeckungspunkt („break-even point“)?“ und
- „Sollte die Akquise nach einer bestimmten Zeit abgebrochen werden, weil der Gewinn nicht weiter zunimmt?“

beantwortet werden.

Optimierungsprobleme des einfachen Akquiseplans

Ein einfacher Akquiseplan gibt die Akquiseaktionen, die darauf folgenden möglichen Kunden-Reaktionen und die nach jeder Kunden-Reaktion zu wählende Akquiseaktion vor und bietet daher keinen Ansatzpunkt für die Optimierung irgendwelcher Akquise-relevanter Aktionen. Das ändert sich durch die Einführung von Kapazitätsschranken für die Akquiseaktionen, deren adäquates, das heißt wirklichkeitsnahes Modell eine unscharfe Kapazitätsschranke ist (Def. 4.2). Bei vorhandenen Kapazitätsschranken stellt sich die Frage nach der auslastungsoptimalen Beschickung des Akquiseplans, die in drei Optimierungsproblemen formuliert wird (siehe Abschnitt 3.4):

1. Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan gleichzeitig unter Einhaltung der Kapazitätsschranken der Akquiseaktionen zugeführt werden kann?
2. Wie groß ist die maximale Anzahl an Kunden, die einem Akquiseplan innerhalb eines gegebenen Zeitraums unter Einhaltung der Kapazitätsschranken für Akquiseaktionen zugeführt werden kann, und wie ist sie über den Zeitraum verteilt?

2. siehe Fußnote 1 auf Seite 347

3. Wie muss eine vorgegebene Anzahl von Kunden einem Akquiseplan unter Einhaltung der Kapazitätsschranken zugeführt werden, damit der dafür benötigte Zeitraum möglichst kurz ist?

Am wichtigsten ist das zweite Optimierungsproblem, denn das erste stellt nur einen Sonderfall des zweiten dar, und das dritte kann durch geeignetes iteratives Lösen des zweiten Optimierungsproblems gelöst werden (Alg. 4.5). Das zweite Optimierungsproblem steht im Mittelpunkt der Betrachtungen in den Abschnitten 4.2 – 4.3. Zuvor wird gezeigt, wie das aufgrund der unscharfen Kapazitätsschranken unscharfe Optimierungsproblem *geschärft* werden kann: Für einen festgelegten Wahrheitswert kann über die *verallgemeinerte Umkehrabbildung der Zugehörigkeitsfunktion* der unscharfen Kapazitätsschranke (Def. 4.3) ein scharfer Wert für die Kapazitätsschranke bestimmt werden, mit dem dann die Optimierung durchgeführt wird. In Anlehnung an die Begrifflichkeit in der stochastischen Optimierung wird vom *scharfen Äquivalent* des unscharfen Optimierungsproblems gesprochen (Def. 4.4). Seine Bedeutung erschließt sich über den für die Schärfung festgelegten Wahrheitswert: Die Einhaltung der resultierenden Kapazitätsschranken ist mit diesem Wahrheitswert *möglich*.

Zur Lösung des zweiten Optimierungsproblems werden vier Ansätze untersucht, die sich in der Bestimmung des deterministischen Äquivalents der stochastischen Nebenbedingungen unterscheiden. Diese stochastischen Nebenbedingungen beschreiben die zufälligen Reaktionen der Kunden, die mit einer Akquiseaktion bearbeitet worden sind.

1. Der *Erwartungswertansatz* und die *Optimierung mit zufälligen Nebenbedingungen* („chance constrained programming“) im Abschnitt 4.2 nutzen die Struktur des Akquiseplans nicht aus. Diese Ansätze werden untersucht, aber nicht weiterverfolgt. Mit dem Erwartungswertansatz sind die Probleme einfach, spiegeln aber – wie alle auf dem Erwartungswert beruhenden Ansätze – die Stochastik der Zufallsgröße nur unzureichend wieder. Der Ansatz als *Optimierungsproblem mit zufälligen Nebenbedingungen* führt zu komplexen nicht linearen Problemen, die mit einem LP-Löser nicht lösbar sind.
2. Der dritte und der vierte Ansatz im Abschnitt 4.3 nutzen die Struktur des Akquiseplans aus, sind *szenarienbasiert* und werden nachfolgend beschrieben.

Bei mehrfacher Zuführung von Kunden überlagern sich gleichsam mehrere einfache Akquisepläne. Die „Kundenflüsse“ beeinflussen sich nicht direkt gegenseitig, aber indirekt durch die gemeinsamen Kapazitätsschranken der Akquiseaktionen. Der einfache Akquiseplan wird zum *zeitlichen Akquiseplan* (Def. 4.26) erweitert. In den Knoten dieses Graphen gelten

- für jeden Zuführungszeitpunkt eigene Fluss-Erhaltungs-Nebenbedingungen und
- für alle Zuführungszeitpunkte gemeinsame Knoten-Kapazitäts-Nebenbedingungen.

Formal liegt ein *Mehrgüterfluss-Problem* mit den zuletzt genannten Nebenbedingungen als *Bündel-Nebenbedingungen* vor. Ohne Bündel-Nebenbedingungen sind die Optimierungsvariablen stets ganzzahlig. Durch Beweis der *totalen Unimodularität der Problematrix* wird gezeigt, dass die Lösungen auch mit den Bündel-Nebenbedingungen stets ganzzahlig sind (Satz 4.13). Diese Eigenschaft geht verloren, wenn die stochastischen *Kunden-Verteilungs-Nebenbedingungen* berücksichtigt werden.

Die stochastischen Kunden-Verteilungs-Nebenbedingungen folgen dem *stochastischen Prozess der Kundenverteilung* (Def. 4.18). *Szenarien* sind Realisierungen dieses stochastischen

Prozesses. Somit ergibt jedes Szenario eine Menge von Kunden-Verteilungs-Nebenbedingungen. Würden alle diese Nebenbedingungen zugleich in ein lineares Optimierungsproblem aufgenommen, dann würde das Ergebnis dem schlechtesten Fall („worst case“) entsprechen, das heißt aus allen Szenarien würde jene Kunden-Verteilungs-Nebenbedingungen wirken, die den „Kundenfluss“ am stärksten einschränken. Um diesen Effekt zu vermeiden, werden zwei andere Ansätze untersucht. Sie unterscheiden sich in der Art und Weise, wie diese Mengen in das Optimierungsproblem eingehen:

1. Beim *Mittelwert-Verfahren* (Abschnitt 4.3.4.4) wird für jedes Szenario ein lineares Optimierungsproblem (das Mehrgüterfluss-Problem mit den Kunden-Verteilungs-Nebenbedingungen des Szenarios) gelöst. Abschließend wird durch Mittelwertbildung der Werte der Optimierungsvariablen über die Szenarien eine Lösung berechnet.
2. Beim erweiterten *Lagrange-Verfahren* (Abschnitt 4.3.4.5) werden die Kunden-Verteilungs-Nebenbedingungen aller Szenarien in die Zielfunktion des (dann quadratischen) Optimierungsproblems aufgenommen und somit bei der Optimierung zugleich berücksichtigt. Der Algorithmus (Alg. 4.6) wurde in der Programmiersprache C++³⁾ implementiert (Abb. 4.15). Die quadratischen Optimierungsprobleme in den Iterationen des Verfahrens werden mit *CPLEX*³⁾ gelöst.

Eine obere Schranke der optimalen Lösungen bildet das (stets ganzzahlige) Optimierungsergebnis des Mehrgüterfluss-Problems *ohne* Kunden-Verteilungs-Nebenbedingungen. Darum können die in der Regel nicht-ganzzahligen Ergebnisse des Mittelwert-Verfahrens und des Lagrange-Verfahrens aufgerundet werden, ohne einen Fluss im Graphen des zeitlichen Akquiseplans zu erzeugen, der die Kapazitätsschranken verletzen würde.

Im Abschnitt 4.3.5 wird die Erzeugung der Szenarien aus Pseudo-Zufallszahlen beschrieben. Ihr Kern ist die Erzeugung von Realisierungen von Zufallsvektoren mit *normierter Multinomialverteilung* (Def. 4.17).

Zur Beurteilung von Mehrgüterfluss-Problem ohne und mit Kunden-Verteilungs-Nebenbedingungen, insbesondere des Mittelwert-Verfahrens und des Lagrange-Verfahrens wird im Abschnitt 4.4 ein Fallbeispiel betrachtet. Alle Verfahren wurden in einem einheitlichen C++-Programm³⁾ implementiert, dessen Funktion über Parameter gesteuert wird. Dieses Programm, seine Ein- und Ausgabedaten und -dateien werden im Abschnitt 4.4.1 genau beschrieben. Das Fallbeispiel beruht auf dem Akquiseplan eines Experten (Abb. 3.5) und den in den Definitionsdateien *.tme und *.tms festgelegten zeitlichen Punktwahrscheinlichkeiten und Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten, die auch bei der Simulation des Kundenverhaltens im Kapitel 3 verwendet werden.

Zur Untersuchung verschiedener Aspekte des Fallbeispiels wird ein *Untersuchungsprogramm* mit 15 Fällen festgelegt (Tabelle 4.2). Untersucht werden:

- Das Mehrgüterfluss-Problem ohne Kunden-Verteilungs-Nebenbedingungen: ein lineares Programm, das sehr schnell mit dem Simplexverfahren gelöst wird und aufgrund der im Abschnitt 4.3.4.3 gezeigten Eigenschaften stets ganzzahlige Lösungen besitzt. Seine

3. siehe Fußnote 1 auf Seite 347

Lösung ist der unter den gegebenen Kapazitätsschranken maximal mögliche „Kundenfluss“ des zeitlichen Akquiseplans.

- Das zweite Optimierungsproblem und, als Sonderfall desselben, das erste Optimierungsproblem mit Kunden-Verteilungs-Nebenbedingungen: Durchgeführt werden Optimierungen mit dem Mittelwert-Verfahren und mit dem erweiterten Lagrange-Verfahren. Beide werden bezüglich ihres Laufzeitverhaltens und der Optimierungsergebnisse bewertet und miteinander verglichen:
- Alle Ergebnisse werden durch die verfügbare Hardware beschränkt: Desktop mit AMD Athlon und Laptop mit Intel Pentium 4 Mobile-Prozessor mit 1,47GHz bzw. 1,8GHz und 1GB bzw. 750MB Arbeitsspeicher sowie 32-bit-Betriebssystem *Microsoft Windows XP*⁴⁾. Besonders der Arbeitsspeicher stellt eine Schranke dar, die ab einer bestimmten Problemgröße überschritten wird und zum Auslagern des Arbeitsspeicherinhalts auf die Festplatte (so genanntes „swappen“) führt, was die Rechenzeit erhöht. Mit einem schnelleren Prozessor, größerem Arbeitsspeicher, einem 64-bit-Betriebssystem und der 64-bit-Version von *CPLEX*⁵⁾ ist eine Leistungssteigerung zu erwarten.
- Außer der Hardware stellt auch der *CPLEX*⁵⁾-Löser eine Beschränkung dar: Beim erweiterten Lagrange-Verfahren traten ab einer bestimmten Anzahl von Nebenbedingungen in der quadratischen Zielfunktion Ausnahmen („exceptions“) auf⁶⁾.
- Mit dem Mittelwert-Verfahren können wirklichkeitsnahe Probleme mit einigen hundert Szenarien und langer Dauer der Kundenzuführung sehr schnell gelöst werden. Die Optimierungsergebnisse sind die Kunden-Anzahlen, die mit der ersten Akquiseaktion innerhalb des Zuführungszeitraums bearbeitet werden können, ohne „im Mittel“ die Kapazitätsschranken für die Akquiseaktionen zu verletzen. Ab etwa 10 Szenarien ändert sich das Optimierungsergebnis nur noch wenig, so dass eine sehr große Anzahl von Szenarien gar nicht notwendig ist.
- Mit dem Lagrange-Verfahren beträgt die Laufzeit stets ein Vielfaches der Rechenzeit, die das Mittelwert-Verfahren benötigt. Beim Sonderfall einmaliger Kundenzuführung treten auch bei 20 Szenarien und einer Akquisedauer von 75 Zeitschritten (Tagen) keine der oben genannten Probleme auf. Die Werte der Optimierungsvariablen sind mit dem Lagrange-Verfahren stets kleiner als mit dem Mittelwert-Verfahren, das heißt zu den einzelnen Zeitschritten im Zuführungszeitraum können weniger Kunden zugeführt werden, wenn die Kapazitätsschranken nicht verletzt werden sollen. Dieses Resultat ist verständlich, weil das Lagrange-Verfahren alle Szenarien zugleich berücksichtigt. Mit zunehmender Anzahl an Szenarien werden die Werte der Optimierungsvariablen kleiner. Somit ist dieses Verfahren bezüglich der Einhaltung der Kapazitätsschranken „sicherer“ als das Mittelwert-Verfahren.
- Bei beiden Verfahren zeigt sich, dass die Rechenzeit durch ein Laufzeit-effizienteres Verfahren zur Erzeugung der Szenarien weiter reduziert werden könnte.

4. siehe auch die Angaben zur Hardware im Abschnitt 4.4.2

5. siehe Fußnote 1 auf Seite 347

6. Das Problem wird in den Anmerkungen zu den Fällen 8 und 9 im Abschnitt 4.4.2.3 genau beschrieben.

Vom einfachen zum dynamischen Akquiseplan

Im Kapitel 5 werden die Beschränkungen des *einfachen* Akquiseplans aufgezeigt.

Mit einem einfachen Akquiseplan werden für alle zu bearbeitenden Kunden dieselben Aktionen angewandt bzw. für alle Kunden mit gleicher Reaktion auf eine Aktion wird eine Folgeaktion festgelegt. Sofern die *Kundendatensatz-Menge* (Def. 5.1) bezüglich des zu bewerbenden Produktes bzw. der zu bewerbenden Dienstleistung homogen ist, ist das Modell einer Akquise in Gestalt des einfachen Akquiseplans ausreichend. Werden jedoch unterschiedliche *Kundensegmente* bearbeitet, so sind eine vorangehende *Kundensegmentierung* und die Durchführung der Akquise mit einem eigenen Akquiseplan für jedes Kundensegment notwendig (Abb. 5.1). Zur Vereinheitlichung von Kundensegmentierung und Akquise wird im Abschnitt 5.2 ein *erweiterter Akquiseplan* eingeführt (Abb. 5.2 und 5.3). Seine Idee besteht darin, mit den Kunden-Reaktionen auch bestimmte, vorab festgelegte Kundenmerkmale (Def. 5.1) zu erfassen. Gegenüber dem einfachen Akquiseplan ist der erweiterte Akquiseplan sehr „aufgebläht“ und praktisch schwer handhabbar, da Aktionen nicht nur für die Reaktionen der Kunden, sondern auch für die sich aus der Reaktion ergebenden Kundenmerkmale festgelegt werden müssen.

Um die genannten Probleme mit dem erweiterten Akquiseplan zu vermeiden und dem Benutzer auch ein Werkzeug für die *Entscheidungsfindung* an die Hand zu geben, wird der *dynamische Akquiseplan* eingeführt (Abb. 5.7). Er modelliert die Akquise als mehrstufigen stochastischen Entscheidungsprozess auf der Grundlage eines Zustandsraumes, dessen Zustände Kundensegmente sind. Sie sind über vorab festgelegte Merkmale definiert, deren Werte auch unbekannt sein können, so dass der bei einer Akquise übliche Informationsgewinn über Kunden – durch eigene Auskünfte der Kunden über sich oder Eindrücke und Erkenntnisse des Akquisiteurs – abgebildet werden kann. Der Motivation des dynamischen Akquiseplans im Abschnitt 5.3.1 folgt im Abschnitt 5.3.2 seine genaue Definition (Def. 5.9), die auf folgenden Begriffen beruht:

- *Kundendatensätze*, das sind Tupel von Werten für Kunden-*Merkmalsvariablen* (Def. 5.1).
- Die Menge aller möglichen Kundendatensätze bildet den *Akquise-Zustandsraum*, die Kundendatensätze seine *Zustände* (Def. 5.2).
- Akquiseaktionen werden in der *Akquise-Entscheidungsmenge* zusammengefasst (Def. 5.3).
- Für einen bestimmten Zustand (das heißt ein Kundensegment) und eine darauf (auf alle Kunden des Kundensegments) angewandte Akquise-Entscheidung ist durch *Akquise-Übergangswahrscheinlichkeiten* der *Akquise-Zustandsübergang* gegeben (Def. 5.4).
- Während beim einfachen Akquiseplan das Ende der Akquise durch die zeitlichen Punktwahrscheinlichkeiten festgelegt ist, ist es beim dynamischen Akquiseplan durch Erreichen eines entweder *positiven* oder *negativen Akquise-Endzustandes* bestimmt (Def. 5.5). Gehört der Datensatz eines Kunden zu einem solchen Zustand, dann hat der Kunde entweder „gekauft“ oder er hat sich endgültig gegen einen „Kauf“ entschieden.

Die wichtigste Erweiterung gegenüber dem einfachen Akquiseplan ist die *stationäre Akquise-Entscheidungsregel* oder *stationäre Politik* (Def. 5.8), die jedem Zustand (Kundensegment) eine Akquiseaktion zuordnet, mit der die Kunden des Segments zu bearbeiten sind. Daraus folgt die nahe liegende Frage nach der *besten* Akquise-Entscheidungsregel oder der *optimalen*

Politik. Um schließlich ein Optimierungsproblem formulieren zu können, werden eine „Zielfunktion“ und „Nebenbedingungen“ eingeführt:

- Eine *Ertragsfunktion* bewertet das Ergebnis der Akquise für jeden positiven Akquise-Endzustand. Das *unscharfe Akquise-Ziel* ist eine Darstellungsform der Ertragsfunktion, die eine Kompensation von Akquise-Ziel und Nebenbedingungen ermöglicht (Def. 5.6).
- Mit den *unscharfen Akquise-Nebenbedingungen* wird jede Akquiseaktion bezüglich ihrer Eignung bei Anwendung in einem bestimmten Zustand (das heißt Anwendung auf ein bestimmtes Kundensegment) bewertet (Def. 5.7). Diese Art der Bewertung erlaubt sowohl die Messung der Eignung in Form von Kosten der Akquiseaktion als auch die subjektive Bewertung durch den Akquisiteur.

Mit dieser Definition des dynamischen Akquiseplans lautet das in Abb. 5.7 illustrierte Optimierungsproblem:

Berechne die Akquise-Entscheidungsregel eines gegebenen dynamischen Akquiseplans so, dass das zu erwartende unscharfe Akquiseziel unter Beachtung der unscharfen Akquise-Nebenbedingungen maximal wird.

Optimierungsprobleme des dynamischen Akquiseplans

Für die Lösung des oben beschriebenen Optimierungsproblems werden im Kapitel 6 exakte Grundlagen gelegt und Verfahren entwickelt. Dazu gehören die Einführung der Grundbegriffe der *dynamischen Optimierung* nach Bellman [9] im Abschnitt 6.1, das *Optimalitätsprinzip* (Satz 6.1) sowie das Prinzip der *Rückwärtsiteration*, das für feste Schrittzahl und deterministischen sowie stochastischen Zustandsübergang beschrieben wird (Sätze 6.2 – 6.3).

Die oben eingeführten Konzepte *unscharfes Ziel* und *unscharfe Nebenbedingung* werden in der neuen Idee der *unscharfen Entscheidung* (Def. 6.6) von Bellman und Zadeh [11] zusammengeführt (Abb. 6.1). Sie ist wiederum die Grundlage für die Definition einer *optimalen Politik für unscharfe Entscheidungsprozesse* (Def. 6.8). Eine genaue Ausarbeitung der Methodik zur (stochastischen) dynamischen Optimierung *mehrstufiger Entscheidungsprozesse* mit fester Stufenzahl folgt im Abschnitt 6.3. Als Hauptaussage hält Satz 6.4 fest, dass die optimale Politik für mehrstufige stochastische Entscheidungsprozesse in unscharfer Umgebung mit fest vorgegebener Prozess-Endzeit (Def. 6.9) weitgehend analog zum scharfen Fall über eine Rückwärtsiteration berechnet werden kann.

Eine völlig neue Problemstellung entsteht durch *implizite* Vorgabe der Prozess-Endzeit, wie beim dynamischen Akquiseplan: Der Entscheidungsprozess ist nicht nach einer festen Stufenzahl beendet, sondern sobald ein *Akquise-Endzustand* (Def. 5.5) erreicht ist (siehe den vorigen Abschnitt). Für die Optimierung eines *mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit* (Def. 6.10) reicht die von Bellman und Zadeh eingeführte Methodik nicht aus. Die Entwicklung eines geeigneten Verfahrens ist der Hauptgegenstand von Kapitel 6 und wird anhand von vier Fragen verfolgt. Sie sind nachfolgend zusammen mit ihren Antworten aufgeführt.

1. Gibt es eine Politik π , die das System für jeden Anfangszustand mit Wahrscheinlichkeit eins in einen Endzustand überführt? Wodurch zeichnet sie sich aus?

Zunächst werden im Abschnitt 6.4.2 die gesuchte Politik als *zulässige Politik* charakterisiert (Def. 6.12) und die Bedingungen ihrer Existenz untersucht. Dabei wird gezeigt, dass die Zulässigkeit einer Politik allein von einer leicht zu überprüfenden Eigenschaft ihrer *Übergangsmatrix* abhängt (Satz 6.6).

2. Wie kann zu einer gegebenen Politik π der Entscheidungsvektor $\mu_D(\pi)$ berechnet werden, das heißt wie kann eine Politik bewertet werden?

Es wird gezeigt, dass der Entscheidungsvektor einer Politik die Lösung eines *Fixpunktproblems* ist. Mit Hilfe des *Weissingerschen Fixpunktsatzes* (Satz 6.8) kann für zulässige Politiken bewiesen werden, dass ein eindeutiger Fixpunkt existiert, dieser durch *Iteration* für einen beliebigen Anfangswert berechnet sowie eine Fehlerabschätzung für jeden Iterationsschritt angegeben werden kann (Satz 6.9).

3. Wie wird eine optimale Politik π^{opt} definiert und unter welchen Voraussetzungen existiert sie?

Auf diese Frage wird im Abschnitt 6.4.3 eine *existenzielle* und eine *konstruktive* Antwort gegeben. Die Menge der Politiken bildet eine *Halbordnung* (Def. 6.13), über die für zulässige Politiken die *Optimalität* definiert wird. Unter der Voraussetzung, dass alle Politiken eines mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit zulässig sind, wird gezeigt, dass die Halbordnung der Politiken ein *endlicher vollständiger Verband* ist (Satz 6.16), womit wiederum die Existenz optimaler Politiken gezeigt wird (Satz 6.17).

Eine konstruktive Antwort gibt der Satz 6.20. Es wird gezeigt, dass die unscharfe Entscheidung einer optimalen Politik die Lösung eines *Fixpunktproblems* ist. Sofern alle Politiken zulässig sind, kann der Fixpunkt *iterativ* für einen beliebigen Anfangswert berechnet werden. Wie bei Satz 6.9 bedient sich die Beweisführung des *Weissingerschen Fixpunktsatzes* und ist unabhängig von der zunächst gezeigten Existenz einer optimalen Politik.

4. Wie kann gegebenenfalls π^{opt} berechnet werden?

Diese Frage ist durch die konstruktive Antwort auf Frage 3 beantwortet.

Wie im Ansatz von Bellman und Zadeh werden die obigen Hauptaussagen für die *Minimum-t-Norm* bewiesen. Im Kontext unscharfer Entscheidungen ist die Minimumbildung nicht die einzig geeignete t-Norm. Daher wird die aus [11] übernommene Def. 6.6 der *unscharfen Entscheidung* in Def. 6.7 verallgemeinert. Ein erster Schritt zur Verallgemeinerung obiger Hauptaussagen ist die Formulierung der Transformationen der beiden Fixpunktprobleme für beliebige t-Normen (Def. 6.17). Danach wird gezeigt, dass die Hauptaussagen ihre Gültigkeit behalten, wenn die in der Transformation verwendete t-Norm die *Lipschitzbedingung* (6.91) erfüllt. Damit sind folgende Verallgemeinerungen möglich:

- Satz 6.22 verallgemeinert Satz 6.9 (Berechnung der unscharfen Entscheidung einer zulässigen Politik),
- Satz 6.24 verallgemeinert Satz 6.16 (die Halbordnung der zulässigen Politiken ist ein endlicher vollständiger Verband),
- Satz 6.25 verallgemeinert Satz 6.17 (Existenz optimaler Politiken),
- Satz 6.26 verallgemeinert Satz 6.20 (Iterative Berechnung optimaler Politiken).

Der Abschnitt 6.5.3 befasst sich mit einigen praktischen Fragen dynamischer Akquisepläne. Dabei wird auseinander gesetzt, dass die Annahme der Zulässigkeit aller Politiken eines mehrstufigen stochastischen Entscheidungsprozesses in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit in der Praxis keine Einschränkung darstellt, also wirklichkeitsnah ist. Auch die Möglichkeit, dass gleiche Aktionen mehrfach angewendet werden, lässt sich durch einfache praktische Maßnahmen verhindern.

Die praktische Umsetzbarkeit der Hauptaussagen des Kapitels wird im Abschnitt 6.6 gezeigt. Die in den Hauptaussagen „skizzierten“ Verfahren (z. B. die iterative Berechnung einer optimalen Politik) werden in Algorithmen ausgearbeitet (z. B. Alg. 6.4) und in *Matlab*⁷⁾ programmiert. Zu ihrer Untersuchung werden Fallbeispiele betrachtet, die auf zwei dynamischen Akquiseplänen aufbauen, die in *Excel*-Dateien⁷⁾ dokumentiert sind.

- Der erste, *einfache dynamische Akquiseplan* ist für die praktische Anwendung zu einfach, dafür aber überschaubar und leicht interpretierbar.
- Der zweite, *komplexe dynamische Akquiseplan* würde sich für die praktische Anwendung eignen. Er ist jedoch schwerer zu überschauen und zu interpretieren.

Alle *Matlab*-Programme, ihre Eingabe- und Ausgabedateien sowie die Schnittstellen mit den dynamischen Akquiseplänen in den *Excel*-Dateien werden im Abschnitt 6.6.3 ausführlich beschrieben.

Eine eigenständige Aufgabe stellt die Erstellung der dynamischen Akquisepläne dar. Sie erfordert die Festlegung

- der *Zustandsmenge* durch die *Merkmalsvariablen* und die *Merkmalsmengen* mit ihrer Untermenge der *beendeten Zustände*,
- der Menge der *Akquise-Entscheidungen*,
- der *stochastischen Übergangsmatrizen*,
- der *unscharfen Nebenbedingungen*
- sowie des *unscharfen Zieles*.

Während diese Festlegungen für den einfachen dynamischen Akquiseplan händisch möglich sind, überfordern sie für den komplexen dynamischen Akquiseplan selbst einen Fachmann. Deshalb wird ein Verfahren entwickelt, das unter bestimmten vereinfachenden Annahmen eine *regelbasierte automatische Generierung* der stochastischen Übergangsmatrizen, der unscharfen Nebenbedingungen und des unscharfen Zieles ermöglicht. Es wird in *Clips*⁷⁾ programmiert und zur Erzeugung des komplexen dynamischen Akquiseplans angewendet. Das Verfahren, die *Clips*-Programme, ihre Ein- und Ausgabedateien sowie die Datenflüsse (Abbildungen 6.10, 6.13, 6.16) werden in den Abschnitten 6.6.2.1 – 6.6.2.3 genau beschrieben.

Die Ergebnisse der Untersuchung mit den Fallbeispielen sind durchweg plausibel und bestätigten die Praxistauglichkeit aller Verfahren. Sie sind in den Abschnitten 6.6.4 (Bestimmung der

7. siehe Fußnote 1 auf Seite 347

Zulässigkeit einer Politik) und 6.6.5 (Lösung des Optimierungsproblems) ausführlich dargestellt.

7.2 Ausblick auf mögliche Erweiterungen

Aufgrund der Breite des in der vorliegenden Arbeit behandelten Themas *Unschärfe stochastische Optimierung und Anwendungen im Marketing* werden mit den neuen Ansätzen und Verfahren auch Fragen aufgeworfen, deren Behandlung in diesem Rahmen nicht möglich ist. Die lohnenswertesten davon werden nachfolgend angerissen.

7.2.1 Simulation der Kundenakquisition mit Akquiseplänen

Erweiterung des Akquisemodells mit dem einfachen Akquiseplan

Die vorliegende Modellierung berücksichtigt das zufällige und zeitliche Verhalten der Kundenreaktionen auf eine Akquiseaktion. Für die Bearbeitung der Akquiseaktionen wird davon ausgegangen, dass diese innerhalb eines Zeitschrittes (Tages) möglich ist. Letzteres ist wirklichkeitsnah, wenn Kapazitätsschranken für die Bearbeitung einzelner Aktionen berücksichtigt werden, wie das für die Optimierungsprobleme im Kapitel 4 der Fall ist. Die Akquise-Simulation im Kapitel 3 berücksichtigt die Kapazitätsschranken nicht. Bei Betrachtung eines einzelnen oder nur weniger Kunden ist dies unerheblich. Für eine größere Anzahl zugeführter Kunden und für die mehrfache Zuführung von Kunden ist eine Erweiterung des Modells mit Berücksichtigung der Kapazitätsschranken notwendig.

Auch die Kapazitätsschranken selbst können noch realistischer modelliert werden. Sie werden hier als voneinander unabhängig angenommen, so als ob jede Akquiseaktion von einer ganz bestimmten Person (z. B. einem Mitarbeiter eines Versicherungsunternehmens) mit seinem spezifischen Zeitbudget bearbeitet würde. Insbesondere bei kleinen und mittleren Unternehmen wird jedoch ein Mitarbeiter für mehrere Aktionen verantwortlich sein und beispielsweise neben der Beantwortung von Kundenanfragen auch Angebote schreiben oder im Außendienst Kunden aufsuchen. Diese gegenseitige Abhängigkeit der Kapazitätsschranken für mehrere Akquiseaktionen kann durch Erweiterung der unscharfen Kapazitätsschranken (Def. 4.2) auf *unscharfe Kapazitätsrelationen* modelliert werden. Deren Berücksichtigung in den Optimierungsproblemen aus Kapitel 4 bedarf weiterer Überlegungen.

Erweiterung des Umfangs simulierbarer ökonomischer Kenngrößen

Neben dem zeitlichen Fluss der Kunden durch den Akquiseplan wird mit der im Kapitel 3 beschriebenen Simulation der zeitliche Verlauf ökonomischer Kenngrößen simuliert. Außer den in Tabelle 3.1 aufgeführten Größen gibt es weitere, die bei der Durchführung einer Kundenakquise oder Marketing-Kampagne interessant sein können. Für einen vorgegebenen Akquiseplan interessiert z. B., mit welchem Anteil die einzelnen Aktionen zum Gesamtertrag beitragen. Um diese festzustellen, könnten die Wege im Akquiseplan, die durch die betreffende Aktion führen und entweder in der positiven oder in der negativen Senke enden, ausgewertet werden. So ließe sich erkennen, ob bestimmte Akquiseaktionen ab einem gewissen Zeitpunkt nicht mehr angewendet werden sollten oder sogar gänzlich überflüssig sind.

7.2.2 Optimale Beschickung eines Akquiseplans

Verbesserung des Lagrange-Verfahrens

Bei dem zur Lösung des unscharfen Mehrgüterfluss-Problems mit stochastischen Nebenbedingungen eingesetzten Lagrange-Verfahren treten ab einer bestimmten Problemgröße Programm-*Ausnahmen* („exceptions“) auf, auf die im Abschnitt 7.1 hingewiesen wird. Sie treten auf, wenn das mit den Klassen der *CPLEX C++*-Schnittstelle⁸⁾ modellierte Problem nach *CPLEX* exportiert wird. Bei Beibehaltung des verwendeten *CPLEX*-Lösers wäre daher Folgendes zu untersuchen:

- Treten die Probleme auch bei leistungsfähigerer Hardware (größerer Arbeitsspeicher) auf?
- Lassen sich die Probleme vermeiden bzw. lässt sich die Problemgröße, bei der sie auftreten, erhöhen, indem vor dem Export in die „*CPLEX*-Welt“ eine geeignete Vorverarbeitung, z. B. das Ausmultiplizieren und Zusammenfassen von Termen vorgenommen wird?

Alternativ könnte auch das Verhalten bei Verwendung eines anderen Lösers für quadratische Optimierungsprobleme untersucht werden.

Effektivere Erzeugung von Szenarien

Beim Mittelwert-Verfahren und Lagrange-Verfahren wird das deterministische Äquivalent des stochastischen Optimierungsproblems mithilfe von *Szenarien* erzeugt. Dabei werden viele Realisierungen von Zufallsvektoren mit Multinomialverteilung berechnet. Dies geschieht über mehrere Zwischenstufen aus gleichverteilten Pseudo-Zufallszahlen, die eine *C++*-Funktion liefert. Laufzeit-Betrachtungen ergaben, dass diese Berechnungen Rechenzeit-intensiv sind. Daher wäre zu untersuchen, ob bei Nutzung der *C++*-Funktion nicht andere Algorithmen bzw. ein alternatives Verfahren ohne Nutzung der *C++*-Funktion schneller wären.

7.2.3 Optimierung mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung

Nutzung von Datenbeständen für die Erstellung von Akquiseplänen

Bei der Erstellung des *komplexen dynamischen Akquiseplans* im Abschnitt 6.6.2 gestaltete sich besonders die Festlegung der *Übergangswahrscheinlichkeiten* schwierig. Auch wenn ein Fachmann mit Erfahrung die Wahrscheinlichkeiten näherungsweise schätzen kann, so ist doch die Anzahl aus Kombinationen von Zustand, Folgezustand und Akquiseaktion geradezu unübersehbar und daher die Festlegung aller Übergangswahrscheinlichkeiten praktisch händisch nicht durchführbar. Eine Alternative zu dem hier entwickelten Verfahren zur automatischen Generierung der Übergangsmatrizen aus einer überschaubaren Anzahl von Angaben ist ihre *Schätzung aus Datenbeständen*. Soll bei Nutzung eines Schätzverfahrens die Qualität der Schätzung zufriedenstellend sein, so ist eine hinreichend umfangreiche Datenmenge nötig sowie eine ausreichende Abdeckung aller Kombinationen aus Zustand, Folgezustand und Akquiseaktion im Datenbestand. Vor der Anwendung des Schätzverfahrens wird bei Echt Daten in der Regel eine Vorverarbeitung nötig sein, um Datenfehler wie Ausreißer zu erkennen und ggf. zu behandeln.

8. siehe Fußnote 1 auf Seite 347

Gerade bei Marketing-Daten ist der Datenumfang sehr groß, ihre Qualität aber stark abhängig von ihrer Erfassung (Sind dafür klare Geschäftsprozesse definiert?) und ihrer Verarbeitung, Speicherung und Konsolidierung. Beispiele solcher Datenquellen werden im Abschnitt 7.3 gegeben.

Für die Simulation von Akquisen mit *einfachen Akquiseplänen* (Kapitel 3) und die darauf aufbauenden Optimierungsprobleme (Kapitel 4) ist die Festlegung von *zeitlichen Punktwahrscheinlichkeiten* und *Wahrscheinlichkeiten des Wechsels nach Wartezeit im Knoten* notwendig. Wenn auch ihre Anzahl überschaubar ist, so könnte ihre Genauigkeit durch ihre Schätzung aus Datenbeständen erhöht werden. Dabei gilt das im vorigen Abschnitt Gesagte entsprechend.

Erweiterung der Definition unscharfer Entscheidungen

Unscharfe Entscheidungen sind der Bewertungsmaßstab für die Optimierung der stationären Politik im Kapitel 6. Sie sind hier zunächst genau so wie in der Arbeit von Bellman und Zadeh [11], in der sie erstmals erwähnt werden, definiert. Mit Def. 6.7 wird eine allgemeinere Definition angegeben und schließlich wird die Definition von Bellman und Zadeh für andere t-Normen als das Minimum definiert. Sowohl die ursprüngliche Definition als auch die angeführten Erweiterungen in dieser Arbeit verknüpfen unscharfe Nebenbedingungen (also unscharfe Mengen) mit t-Normen oder ähnlichen Operatoren. In der scharfen dynamischen Optimierung entsprechen solchen Nebenbedingungen häufig *Kosten*, die sich im Verlauf des Entscheidungsprozesses addieren. Ihre Summe wird schließlich von dem *Ertrag*, der beim Erreichen eines Endzustands anfällt, abgezogen. Diese Betrachtungsweise entspricht sehr gut den Verhältnissen bei einer Kundenakquise. Bei der unscharfen Entscheidung findet aber die gewohnte Addition und Subtraktion nicht statt, sondern „normierte Kosten“ (in Gestalt der Wahrheitswerte der unscharfen Nebenbedingungen) werden mit t-Normen verknüpft und schließlich auch mit dem „normierten Ertrag“ (in Gestalt der Wahrheitswerte des unscharfen Ziels). Durch die Verknüpfung mit geeigneten t-Normen, z. B. mit dem Produkt, wird gewährleistet, dass mehrfach auftretende Kosten die unscharfe Entscheidung „verschlechtern“ und hohe Kosten den Ertrag stärker „reduzieren“ als geringe Kosten. Ein linearer Zusammenhang besteht aber nicht. Daher wäre zu untersuchen, welche t-Norm für Ertrags- und Kostenbetrachtungen am geeignetsten ist und im Idealfall wäre das „Zusammenfallen“ („confluence“ [11]) von Ziel und Nebenbedingungen so zu definieren, dass

- der scharfe Fall mit additiven Kosten und Subtraktion ihrer Summe vom Ertrag ein Sonderfall des unscharfen Falles ist
- und dennoch die hier gezeigte iterative Berechenbarkeit der optimalen Politik möglich ist.

Flexible Granularität der Merkmalsvariablen in dynamischen Akquiseplänen

Der dynamische Akquiseplan beruht auf einer *Zustandsmenge*, die durch Merkmalsvariablen und Merkmalsmengen gegeben ist. Die fest vorgegebene *Granularität* der Merkmalsvariablen in Verbindung mit den Übergangswahrscheinlichkeiten ist nicht unproblematisch. Bei erstmaliger Durchführung einer Akquise werden die Übergangswahrscheinlichkeiten bei relativ feiner Granularität nur sehr ungenau angegeben werden können. Dagegen wären Erfahrungswerte oder Daumenregeln, auf die in solchen Situationen zurückgegriffen werden muss, vertrauenswürdiger, wenn die Merkmalsvariablen eher grobe Granularität besäßen. Auch bei der Schätzung der Übergangswahrscheinlichkeiten aus Datenbeständen des beobachteten Kunden-

verhaltens, wäre bei empirischen Daten geringen Umfangs die Vertrauenswürdigkeit bei grober Granularität größer als bei einer feinen Granularität. Daraus wird ersichtlich,

- dass die Granularität in der Praxis von der Güte der Übergangswahrscheinlichkeiten abhängen sollte,
- dass die Granularität der Merkmalsvariablen flexibel sein sollte und
- Berechnungen der Erreichbarkeit eines Endzustandes und von optimalen Politiken bei jeder Änderung der Granularität erneut durchgeführt werden müssen, wobei ein Verfahren zu bevorzugen wäre, das dabei auf die bereits erzielten Ergebnisse zurückgreift.

Alternative Lösungsansätze für das Optimierungsproblem

Aus der Literatur ist bekannt, dass dynamische Optimierungsprobleme, die bestimmte Eigenschaften besitzen, auf *lineare Optimierungsprobleme* zurückgeführt werden können [140]. Die Ergebnisse von Hilfssatz 6.15 deuten darauf hin, dass diese Voraussetzungen in Verbindung mit einer geeigneten t-Norm (z. B. dem Produkt) erfüllt sind. Mit der Minimum-t-Norm wäre das entsprechende Optimierungsproblem vermutlich gemischt-ganzzahlig. Dies wäre genauer zu untersuchen. Ein solches lineares Optimierungsproblem würde die unscharfe Entscheidung einer optimalen Politik liefern. Die darauf aufbauende Bestimmung einer optimalen Politik bedürfte der weiteren Untersuchung.

7.2.4 Softwareprodukt-Prototyp

Für die Überprüfung der in der vorliegenden Arbeit entwickelten Modelle und Optimierungsverfahren in der Praxis sollte der Prototyp eines Softwareproduktes entwickelt werden. Dieser Prototyp wäre ein Entscheidungsunterstützungssystem („decision support system“). Das Nebeneinander der einerseits auf Akquiseplänen beruhenden Simulation und Optimierungsprobleme und der andererseits auf dynamischen Akquiseplänen beruhenden dynamischen Optimierung in unscharfer Umgebung wäre kein Makel, da es für beide Ansätze spezifische Anwendungen gibt. Besonders vorteilhaft wäre ein solcher Prototyp allerdings, wenn die Modelle vereinheitlicht oder ineinander überführt werden könnten.

Unter dem Gesichtspunkt der Software-Entwicklung sind für die Entwicklung eines Softwareprodukt-Prototyps noch folgende Arbeiten durchzuführen:

- Alle Verfahren müssen einheitlich in einer Programmiersprache nach denselben Qualitätsrichtlinien programmiert und dokumentiert werden.
- Eine grafische Benutzeroberfläche ist zu entwerfen, zu programmieren und über geeignete Schnittstellen mit den Anwendungsprogrammen zu verbinden.
- Ein Datenmodell zur Ablage aller benötigten Daten ist zu entwickeln sowie Schnittstellen, über die die Anwendungsprogramme und ggf. die grafische Benutzeroberfläche auf die Datenbank zugreifen können.

Ein solcher Prototyp müsste dann in der Praxis getestet werden. Nach eventuell notwendigen Anpassungen könnte er als professionelles Softwareprodukt eingesetzt werden.

7.3 Anwendungsgebiete

Der Ausgangspunkt der vorliegenden Arbeit war das Software-Werkzeug *AkquiSys*, mit dem der Verlauf einer Kundenakquisition geplant, organisiert, durchgeführt und ausgewertet werden kann. Daher können die hier vorgestellten Modelle und Verfahren in den Gebieten eingesetzt werden, in denen schon *AkquiSys* Anwendung findet. Dazu gehören die Kundenakquisition von kleinen und mittleren Banken, Sparkassen und Versicherungsagenturen.

Das Anwendungsgebiet ist jedoch nicht auf solche Unternehmen beschränkt. Grundsätzlich können die Verfahren zur Planung, Organisation und *Optimierung der Kundenakquisition* in beliebigen Wirtschaftszweigen eingesetzt werden. Besonders lohnenswert und zugleich geeignet scheint der Einsatz für Anwendungen, bei denen sehr große Kundenzahlen verarbeitet werden. Lohnenswert, weil die Anforderungen an die Planung, Organisation, Durchführung und Auswertung der Akquise bei sehr großen Kundenzahlen besonders hoch sind. Durch eine optimierte Planung und Durchführung können Kosten gespart und Ressourcen zielgerichteter und wirksamer eingesetzt werden. Geeignet, weil die bei großen Kundenzahlen im Laufe der Zeit entstehenden Datenbestände eine (genaue) Schätzung von Verfahrensgrößen erlauben, die ansonsten durch einen Fachmann händisch festzulegen wären. Große Kundenmengen und Datenbestände fallen z. B. in folgenden Bereichen an:

- Kundenbindungsprogramme des Einzelhandels⁹⁾,
- Vielfliegerprogramme von Fluggesellschaften¹⁰⁾,
- Vielfahrerprogramme von Bahngesellschaften¹¹⁾,
- Treueprogramme von Hotelketten¹²⁾,
- Versandhandel,
- Telefongesellschaften,
- Banken und Versicherungen,
- karitative Organisationen.

Die unscharfe Optimierung ist eine ganze Teildisziplin der Mathematik bzw. der Unternehmensforschung („operations research“) und wird in ganz unterschiedlichen Wissenschaften angewandt, nicht nur in den Wirtschaftswissenschaften, sondern auch in den Ingenieur- und Gesellschaftswissenschaften und in der Medizin. Unter den zahlreichen Anwendungsbeispielen, die Kacprzyk für die unscharfe dynamische Optimierung nennt [86, 87] befinden sich:

- sozioökonomische Regionalplanung,
- Steuerung der Überschwemmungsbekämpfung,
- Forschungs- und Entwicklungsplanung,

9. z. B. *PAYBACK* (<http://www.payback.de>) oder *HappyDigits* (<http://www.happydigits.de>)

10. z. B. *Miles & More* (<http://www.milesandmore.de>) oder *Fréquence Plus* (<http://www.airfrance.fr>)

11. z. B. *bahn.comfort* (<http://www.bahn.de>) oder *ÖBB Club & Bonus* (<http://www.oebb.at>)

12. z. B. *Starwood Preferred Guest* (<http://www.spg.com>) oder *MaS* (<http://www.solmelia.es>)

- Kraftwerkseinsatzplanung,
- operationsbegleitende Narkose-Überwachung,
- Ressourcen-Zuordnung,
- Lagerverwaltung.

Insofern Probleme die Gestalt mehrstufiger stochastischer Entscheidungsprozesse in unscharfer Umgebung mit implizit vorgegebener Prozess-Endzeit besitzen (oder als solche beschrieben werden können), können die Verfahren aus Kapitel 6 zur Berechnung einer optimalen Entscheidungsregel angewendet werden.

Literaturverzeichnis

Die Literaturverweise sind alphabetisch nach Autoren sortiert und für denselben Autor nach Erscheinungsjahr.

- [1] Aigner, M.: Diskrete Mathematik. Vieweg-Verlag, Braunschweig, Wiesbaden, 1993.
- [2] Alles, H. K.; von Estorff, R.: Erfolge und Probleme beim Direct Response Marketing von Finanzdienstleistungen. In: Betsch, O.; Otto, K.-F. (Hrsg.): Vertriebshandbuch für Finanzdienstleistungen. Fritz Knapp Verlag, 1989, S. 113 – 124.
- [3] Aumann, G.: Angewandte rechnergestützte Geometrie. Vorlesungsskriptum, Mathematisches Institut und Institut für Informatik, Technische Universität München, 1989.
- [4] Backhaus, K.: Multivariate Analysemethoden. 8., verb. Aufl., Springer-Verlag, Berlin u. a., 1996.
- [5] Bandemer, H.; Gottwald, S.: Einführung in Fuzzy-Methoden: Theorie und Anwendungen. 4., überarb. u. erw. Aufl., Akademie-Verlag, Berlin, 1993.
- [6] Bandemer, H.: Ratschläge zum mathematischen Umgang mit Ungewißheit. Teubner-Verlag, Stuttgart, Leipzig, 1997.
- [7] Barrett, J. D.; Woodall, W. H.: A Probabilistic Alternative to Fuzzy Logic Controllers. In: IIE Transactions. Bd. 29, 1997, S. 459 – 467.
- [8] Bauer, H.: Wahrscheinlichkeitstheorie. Verlag Walter de Gruyter, Berlin, New York, 2002.
- [9] Bellman, R.: Dynamic Programming. Princeton University Press, New Jersey, USA, 1957.
- [10] Bellman, R. E.; Kalaba, R.: Dynamic Programming and Modern Control Theory. Academic Press, New York, 1965.
- [11] Bellman, R. E., Zadeh, L. A.: Decision-Making in a Fuzzy Environment. In: Management Science. Bd. 17, 1970, Nr. 4, B-141 – B-164.
- [12] Bellman, R.; Giertz, M.: On the Analytic Formalism of the Theory of Fuzzy Sets. In: Information Sciences. Bd. 5, 1973, S. 149 – 156.
- [13] Bellman, R. E., Zadeh, L. A.: Local and Fuzzy Logic. In: Dunn, J. M., Epstein, G. (Hrsg.): Modern Uses of Multiple-value Logic. Kluwer Academic Publishers, Dordrecht, 1977, S. 105 – 151, 158 – 165.
- [14] Benölken, H.: Microsegmentierung: Wem nützt es? In: Versicherungswirtschaft. Nr. 4, 1995, S. 228 – 229.
- [15] Berry, M. J. A.; Linoff, G.: Data Mining Techniques: For Marketing, Sales, and Customer Support. Wiley, New York u. a., 1997.
- [16] Bertsekas, D. P.: Dynamic Programming and Stochastic Control. Academic Press, New York, 1976.
- [17] Bertsekas, D. P.: Linear Network Optimization: Algorithms and Codes. MIT Press, Cambridge u. a., 1991.

-
- [18] Bertsekas, D. P.: Nonlinear Programming. Athena Scientific, Belmont, Massachusetts, 1995.
 - [19] Bezdek, J.: Fuzziness vs. Probability – Again (!?). In: IEEE Transactions on Fuzzy Systems. Bd. 2, 1994, Nr. 1, S. 1 – 3.
 - [20] Bialy, H.; Olbrich, M.: Optimierung. Fachbuchverlag, Leipzig, 1975.
 - [21] Biewer, B.: Fuzzy-Methoden: Praxisrelevante Rechenmodelle und Fuzzy-Programmiersprachen. Springer-Verlag, Berlin u. a., 1997.
 - [22] Birge, J. R.; Louveaux, F.: Introduction to Stochastic Programming. Springer-Verlag, New York, Berlin, Heidelberg, 1997.
 - [23] Birkhoff, G.: Lattice Theory. American Mathematical Society, Providence, Rhode Island, 3. Aufl., 1967.
 - [24] Bissantz, N.; Hagedorn, J.: Data Mining im Controlling. Reihe: Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik), Friedrich-Alexander-Universität Erlangen/Nürnberg. Bd. 29, 1996, Nr. 7.
 - [25] Böhme, G.: Fuzzy-Logik: Einführung in die algebraischen und logischen Grundlagen. Springer-Verlag, Berlin u. a., 1993.
 - [26] Robert Bosch Hausgeräte GmbH, München: Verkaufsprospekt: Waschmaschinen, Wäschetrockner, Vollwaschtrockner. Dezember 1999.
 - [27] Bretthauer, G. u. a.: Eine gute Theorie findet ihre Bestätigung in der Praxis. In: Automatisierungstechnische Praxis (atp). Bd. 40, 1998, Nr. 6, S. 65 – 77.
 - [28] Breymann, U.: C++: eine Einführung. Hanser-Verlag, München, Wien, 5. Aufl., 1999.
 - [29] Buldt, B.: Supervaluierungsfuzzysoritalhistorisch, oder: ein kurzer Bericht der langen Geschichte, wie die Vagheit auf den Begriff und unter die Formel kam. In [145], S. 41 – 85.
 - [30] Bundesverband Finanzdienstleistungen e. V. Berlin: FiANTEC Professional & AkquiSys: Moderne Systeme für Kundengewinnung und provisionierten Vertrieb. In: Finanz Forum & Finanz-Fach-Office: Organ des Bundesverbands Finanzdienstleistungen e. V., Berlin. 1999, No. 1, S. 13.
 - [31] Chang, S. S. L.: Fuzzy Dynamic Programming and the Decision Making Process. In: Proceedings of the Third Princeton Conference on Information Sciences, Princeton, New Jersey, 1969.
 - [32] Cromme, L.; Weber, K.; Sun, Z.: Bericht zur Zusammenarbeit mit Fa. *Format*. Interner Bericht, Lehrstuhl für Numerische und Angewandte Mathematik, Brandenburgische Technische Universität Cottbus, Juni 1999.
 - [33] Cromme, L.; Weber, K.; Sun, Z.: Bericht zum Projekt AkquiSys. Projektbericht, Lehrstuhl für Numerische und Angewandte Mathematik, Brandenburgische Technische Universität Cottbus, Juli 1999.
 - [34] Cromme, L.; Weber, K.; Würll, S.: Einsatz eines Markov-Ketten-Modells zur Optimierung der Kundenakquisition – mit einem Fallbeispiel aus der Versicherungsbranche. Vortrag auf der DMV-Jahrestagung, Mainz, 5. – 11. September 1999.
-

-
- [35] Deiser, O.: Einführung in die Mengenlehre. Springer-Verlag, Berlin u. a., 2002.
- [36] Devroye, L.: Non-Uniform Random Variate Generation. Springer-Verlag, New York u. a., 1986.
- [37] Dubois, D.; Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, Boston u. a., 1980.
- [38] Dubois, D.; Prade, H.; Yager, R. (Hrsg.): Readings in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [39] Dubois, D.; Prade, H.: Advances in the Egalitarian Approach to Decision-Making in a Fuzzy Environment. In: [175], S. 213 – 240.
- [40] Dupacova, J.: Multistage Stochastic Programs: the State-of-the-Art and Selected Bibliography. In: Kybernetika. Bd. 31, 1995, Nr. 2, S. 151 – 174.
- [41] Dupacova, J.; Consigli, G.; Wallace, S. W.: Scenario for Multistage Stochastic Programs. In: Annals of Operations Research. Bd. 100, 2000, S. 25 – 53.
- [42] Dupacova, J.; Gröwe-Kuska, N.; Römisch, W.: Scenario Reduction in Stochastic Programming: An Approach Using Probability Metrics. Preprint 00-09, Institut für Mathematik, Humboldt-Universität zu Berlin, 2000.
- [43] Dupacova, J.; Gröwe-Kuska, N.; Römisch, W.: Constructing Scenario Trees from Simulation Scenarios by Transportation Distances. Manuskript, Humboldt-Universität zu Berlin, Institut für Mathematik; Karls-Universität, Prag, Lehrstuhl für Wahrscheinlichkeitstheorie und mathematische Statistik, 2001.
- [44] Eaton, J. H.; Zadeh, L. A.: Optimal Pursuit Strategies in Discrete-State Probabilistic Systems. In: Transactions of the ASME: Journal of Basic Engineering. Bd. 84, Reihe D, 1962, Nr. 1, S. 23 – 29.
- [45] Einstein, A.: Geometrie und Erfahrung. Sitzungsberichte der Preußischen Akademie der Wissenschaften, 1921, SB I, S. 123 – 130. Wiederabdruck in: Akademie der Wissenschaften der DDR (Hrsg.): Albert Einstein: Akademie-Vorträge. Akademie-Verlag, Berlin, 1979.
- [46] Elkan, C.: The Paradoxical Success of Fuzzy Logic. In: IEEE Expert. August 1994, S. 3 – 8. Mit 15 Antworten auf den Seiten 9 – 46. Erstveröffentlichung unter demselben Titel in: Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington, D.C., Juli 1993, MIT Press, S. 698 – 703.
- [47] Elkan, C.: The Paradoxical Controversy over Fuzzy Logic. In: IEEE Expert. August 1994, S. 47 – 49.
- [48] Engels, F.: Herrn Eugen Dührings Umwälzung der Wissenschaft („Anti-Dühring“). In: Institut für Marxismus-Leninismus beim ZK der SED (Hrsg.): Karl Marx, Friedrich Engels: Ausgewählte Werke in sechs Bänden. Band V, Dietz Verlag Berlin, 1986.
- [49] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery: An Overview. In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. (Hrsg.): Advances in Knowledge Discovery and Data Mining. MIT Press, Menlo Park, California, 1996.
-

-
- [50] Felger, M.: Beschreibung des Fahrverhaltens eines Autofahrers im „Stop-and-Go,-Verkehr mit Hilfe eines neuronalen Fuzzy-Systems. Diplomarbeit, Universität Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz, 21.07.1994.
 - [51] Finnie, G., Sun, Z., Weber, K.: A Multiagent-Based Intelligent Broker Architecture for Bargaining Processes. In: Proceedings of The Australian Workshop on AI in Electronic Commerce, Sydney, 06.12.1999, 47 – 56.
 - [52] Finnie, G., Sun, Z., Weber, K.: An Intelligent Broker-Centered Multiagent Architecture for Bargaining Processes. In: Proceedings of The 2000 International Conference on Artificial Intelligence, Las Vegas, 26.-29.06.2000.
 - [53] Finnie, G., Sun, Z., Weber, K.: Multiagent-Based Electronic Bargaining Processes. In: Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA'2000), Wollongong, Australien, 11.-13.12.2000.
 - [54] Fishman, G. S.: Monte Carlo – Concepts, Algorithms, and Applications. Springer-Verlag, New York, Berlin, Heidelberg, 1996.
 - [55] Fletcher, R.: Practical Methods of Optimization. Wiley, Chichester u. a., 2. Aufl., 1987.
 - [56] Föllinger, O.: Regelungstechnik. Hüthig-Verlag, Heidelberg, 6. Aufl., 1990.
 - [57] Format Technology GmbH: Handbuch für AkquiSys. <http://www.format.bic-ffo.de>, 11.08.2000¹⁾.
 - [58] Fuzzy logic mailing list der Technischen Universität Wien, Österreich. <http://www.dbai.tuwien.ac.at/marchives/fuzzy-mail/index.html>, November 1999.
 - [59] Gentle, J. E.: Random Number Generation and Monte Carlo Methods. Springer-Verlag, New York, Berlin, Heidelberg, 1998.
 - [60] Giarratano, J. C.: CLIPS User's Guide, Version 6.20. 31.03.2002. <http://www.ghg.net/clips/CLIPS.html>.
 - [61] Goethe, J. W. v.: Faust. Insel-Verlag, Leipzig, 1958.
 - [62] Gönül, F.; Ze Shi, M.: Optimal Mailing of Catalogs: A New Methodology Using Estimable Structural Dynamic Programming Models. In: Management Science. Bd. 44, 1998, Nr. 9, S. 1249 – 1262.
 - [63] Goos, G.: Vorlesungen über Informatik. Band 1: Grundlagen und funktionales Programmieren. Springer-Verlag, Berlin u. a., 2. Aufl., 1997.
 - [64] Goos, G.: Vorlesungen über Informatik. Band 4: Paralleles Rechnen und nicht-analytische Lösungsverfahren. Springer-Verlag, Berlin u. a., 1998.
 - [65] Gottwald, S.: Fuzzy Sets and Fuzzy Logic: Foundations of Application from a Mathematical Viewpoint. Vieweg, Braunschweig, Wiesbaden, 1993.
 - [66] Gottwald, S.: Axiomatizations of t-Norm Based Logics – A Survey. In: Soft Computing. Bd. 4, 2000, S. 63 – 67.

1. Die Seite besteht nicht mehr. Das Unternehmen heißt jetzt *ELAXY Format GmbH* und ist ein Teil der *ELAXY GmbH*, Coburg; <http://www.elaxy.de>.

-
- [67] Gröwe-Kuska, N.; Heitsch, H.; Römisch, W.: Modellierung Stochastischer Datenprozesse für Optimierungsmodelle der Energiewirtschaft. In: Tagungsband: IT-Lösungen für die Energiewirtschaft in liberalisierten Märkten, Leverkusen, 26.-27.11.2001. VDI-Berichte Bd. 1647, VDI-Verlag, Düsseldorf, 2001, S. 69 – 78.
- [68] Hájek, P.; Paris, J.: A Dialogue on Fuzzy Logic. In: Soft Computing. Bd. 1, 1997, Nr. 1, S. 3 – 5.
- [69] Hartung, J.; Elpelt, B.; Klösener, K.-H.: Statistik. Oldenbourg-Verlag, München, Wien, 12. Aufl., 1999.
- [70] Hermes, H.: Einführung in die Verbandstheorie. Springer-Verlag Berlin, Heidelberg, New York, 2. Aufl., 1967.
- [71] Heuser, H.: Lehrbuch der Analysis Teil 1, Teubner-Verlag, Stuttgart, 2. Aufl., 1986.
- [72] Heuser, H.: Funktionalanalysis: Theorie und Anwendungen. Teubner-Verlag, Stuttgart, 2. Aufl., 1986.
- [73] Hilke, W.: Dienstleistungs-Marketing: Banken und Versicherungen – Freie Berufe – Handel und Transport – Nicht-erwerbswirtschaftliche orientierte Organisationen. Schriften zur Unternehmensführung. Bd. 35, Gabler-Verlag, Wiesbaden, 1989.
- [74] Hinderer, K.: Grundbegriffe der Wahrscheinlichkeitstheorie. Springer-Verlag, Berlin u. a., 3. Aufl., 1985.
- [75] Hisdal, E.: Are Grades of Membership Probabilities? In: Fuzzy Sets and Systems. Bd. 25, 1988, S. 325 – 348.
- [76] Holmblad, L. P.; Østergaard, J.-J.: Control of a Cement Kiln by Fuzzy Logic. In: Gupta, M. M. Sanches, E. (Hrsg.): Fuzzy Information and Decision Processes. North-Holland, Amsterdam u. a., 1982, S. 289 – 399.
- [77] Höppner, F.; Klawonn, F., Kruse, R.: Fuzzy-Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse. Vieweg-Verlag, Braunschweig, Wiesbaden, 1997.
- [78] ILOG: ILOG CPLEX 8.1 Reference Manual. Dezember 2002.
- [79] ILOG-Website: ILOG CPLEX FAQ. <https://support.ilog.fr/private/products/cplex/faq.cfm>, 28.12.2004.
- [80] Iwamoto, S.; Fujita, T.: Stochastic Decision-Making in a Fuzzy Environment. In: Journal of the Operations Research Society of Japan. Bd. 38, 1995, Nr. 4, S. 467 – 482.
- [81] Iwamoto, S.: Fuzzy Dynamic Programming in the Stochastic Environment. In: [175], S. 27 – 52.
- [82] Jordan, Z.: The Development of Mathematical Logic in Poland Between the Two Wars. In: [109].
- [83] Jonsbraten, T. W.; Wets, R. J.-B.; Woodruff, L.: A Class of Stochastic Programs with Decision Dependent Random Elements. In: Annals of Operations Research. Vol. 82, 1998, S. 83 – 106.
- [84] Kacprzyk, J.: Multistage Decision-Making under Fuzziness. Theory and applications. Verlag TÜV Rheinland, Köln, 1983.
-

-
- [85] Kacprzyk, J., Esogbue, A. O.: Fuzzy Dynamic Programming: Main developments and applications. In: Fuzzy Sets and Systems. Bd. 81, 1996, S. 31 – 45.
 - [86] Kacprzyk, J.: Multistage Fuzzy Control. Wiley, Chichester, 1997.
 - [87] Kacprzyk, J., Esogbue, A. O.: Fuzzy Dynamic Programming: Basic Issues and Problem Classes. In: [175], S. 1 – 26.
 - [88] Kall, P.; Wallace, S. W.: Stochastic Programming. Wiley, Chichester, New York, Brisbane, 1994.
 - [89] Keppler, M.: Wirksame Verkaufsförderung durch psychologische Kundensegmentierung. Thesis, Nr. 2, 1993, S. 20 – 23.
 - [90] Kernighan, B. W.; Ritchie, D. M.: Programmieren in C. Verlag Carl Hanser, München, Wien, 2. Aufl., 1990.
 - [91] Klein Haneveld, W. K.: Duality in Stochastic Linear and Dynamic Programming. Springer-Verlag, Berlin u. a., 1986.
 - [92] Klingsporn, B.: Teilmärkte bilden: Yuppie oder Skippie – wer ist ihr Kunde? Bank Magazin, Nr. 7, 1996, S. 34 – 42.
 - [93] Klir, G. J.; Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River, New Jersey, 1995.
 - [94] Koppelman, U.: Marketing: Einführung in Entscheidungsprobleme des Absatzes und der Beschaffung. Werner Verlag, Düsseldorf, 5. Aufl., 1997.
 - [95] Kruse, R.; Schwecke, E.; Heinsohn, J.: Uncertainty and Vagueness in Knowledge Based Systems: Numerical Methods. Springer-Verlag, 1991.
 - [96] Kulisch, U. W.; Miranker, W. L.: Computer Arithmetic in Theory and Practice. Academic Press, New York u. a., 1981.
 - [97] Küspert, A.: Bildung und Bewertung strategischer Geschäftsfelder. Die Bank, Nr. 8, 1991, S. 425 – 434.
 - [98] Küspert, A.: Kundengruppenbildung im Privatkundengeschäft von Kreditinstituten – eine Fallstudie. Zeitschrift für Bankrecht und Bankwirtschaft, Nr. 3, 1992, S. 184 – 201.
 - [99] Lai, Y.-J.; Hwang, C.-L.: Fuzzy Mathematical Programming – Methods and Applications. Springer-Verlag, Berlin, Heidelberg, 1992.
 - [100] Liberty, J.: Jetzt lerne ich C++. Verlag Markt und Technik, Haar, 1997.
 - [101] Link, J.; Hildebrand, V.: Database Marketing und Computer Aided Selling: Strategische Wettbewerbsvorteile durch neue informationstechnologische Systemkonzepten. Verlag Vahlen, München, 1993.
 - [102] Link, J.; Hildebrand, V.: Verbreitung und Einsatz des Database Marketing und CAS: Kundenorientierte Informationssysteme in deutschen Unternehmen. Verlag Vahlen, München, 1994.
 - [103] Luhandjula, M. K.; Gupta, M. M.: On Fuzzy Stochastic Optimization. In: Fuzzy Sets and Systems. Bd. 81, 1996, S. 47 – 55.
-

-
- [104] Lukasiewicz, J.: I. On the Notion of Possibility. II. On Three-Valued Logic. In: [109], S. 15 – 18. (englische Übersetzung der polnischen Originalveröffentlichung von 1920)
- [105] Lukasiewicz, J.: On Determinism. In: [109], S. 19 – 39. (englische Übersetzung der polnischen Niederschrift einer Rede an der Universität Warschau im Jahre 1922)
- [106] Lukasiewicz, J.: Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. In: *Comptes rendus de séances de la Société de Sciences et des Lettres de Varsovie*. Cl. iii, 23, 1930, S. 51 – 77.
- [107] Mamitzsch, V.: Fuzzy Theorie als Alternative zur Stochastik – Was heißt hier: Eine Alternative? In [145], S. 239 – 242.
- [108] Mayer, J.: Computational Techniques for Probabilistic Constrained Optimization Problems. In: Marti, K.: *Stochastic Optimization – Numerical Methods and Technical Applications*. Proceedings of GAMM/IFIP-Workshop, Neubiberg, 29.-31.05.1990, Springer-Verlag, Berlin u. a., 1992.
- [109] McCall, S. (Hrsg.): *Polish Logic 1920 – 1939*. Clarendon Press, Oxford, 1967.
- [110] McNeill, D.; Freiberger, P.: *Fuzzy Logic: Die „unscharfe“ Logik erobert die Technik*. Droemersch Verlagsgesellschaft Th. Knaur Nachf., München, 1996.
- [111] Meffert, H., Bruhn, M.: *Dienstleistungsmarketing: Grundlagen – Konzepte – Methoden*. Gabler, Wiesbaden, 3. Aufl., 2000.
- [112] Miele & Cie. GmbH & Co., Gütersloh: *Verkaufsprospekt: Elektrogeräte für Haushalt und Kleingewerbe: Programm Übersicht*, Ausgabe: Dezember 1999.
- [113] Miller, A.: *Death of a Salesman*. In: *Arthur Miller's Collected Plays*. The Viking Press, New York, 1960.
- [114] Möller, A., Römisch, W., Weber, K.: A New Approach to O&D Revenue Management Based on Scenario Trees. *Journal of Revenue and Pricing Management*, Bd. 3, 2004, Nr. 3, S. 265 – 276.
- [115] Molodtsov, D.: Soft Set Theory – First Results. *Computers and Mathematics with Applications*, Bd. 37, 1999, S. 19 – 31.
- [116] Müller-Hagedorn, L.: *Einführung in das Marketing*. Wissenschaftliche Buchgesellschaft, Darmstadt, 2. Aufl., 1996.
- [117] Murty, K. G.: *Linear Programming*. Wiley, New York, 1983.
- [118] Murty, K. G.: *Network Programming*, Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [119] Nauck, D.; Klawonn, F.; Kruse, R.: *Neuronale Netze und Fuzzy-Systeme*. Vieweg-Verlag, Braunschweig, Wiesbaden, 2. Aufl., 1996.
- [120] Nemhauser, G. L.; Wolsey, L. A.: *Integer and Combinatorial Optimization*. Wiley, New York et. al., 1988.
- [121] Nocedal, J.; Wright, J. W.: *Numerical Optimization*. Springer-Verlag, New York, Berlin, Heidelberg, 1999.
-

-
- [122] Nwama, H. S; Ndumu, D. T.: An Introduction to Agent Technology. In: Nwana, H. S., Azarmi, N. (Hrsg.): Software Agents and Soft Computing: Towards Enhancing Machine Intelligence. Springer-Verlag, Berlin, Heidelberg, New York, 1997, S. 3 – 26.
 - [123] Nieraad, C.: Marketingstrategien für den Finanzdienstleistungsmarkt: Eine integrierte Wettbewerbs- und Bedarfsanalyse. Gabler-Verlag, Wiesbaden, 1994.
 - [124] Papageorgiou, M.: Optimierung, Oldenbourg-Verlag, München, 1991.
 - [125] Pawlak, Z.: Rough Sets – Basic Concepts. Warsaw University of Technology, Institute of Computer Science, Research report 13/92, Warschau, 1992.
 - [126] Pawlak, Z.: Rough Sets: Present State and Future. Warsaw University of Technology, Institute of Computer Science, Research report 20/93, Warschau, 1993.
 - [127] Peschel, M.: Fractal Logics and Systems Theory. In: Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz (FH) (Hrsg.): Proceedings of Fuzzy 96: Fuzzy Logic in Engineering and Natural Sciences. Zittau, 1996, S. 484 – 492.
 - [128] Peschel, M.: Fractal Logics of Nature. In: Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz (FH) (Hrsg.): Proceedings of Fuzzy 96: Fuzzy Logic in Engineering and Natural Sciences. Zittau, 1996, S. 9 – 18.
 - [129] Peschel, M.: Fractal Logics versus Fuzzy Logics. In: Hochschule Zittau/Görlitz (FH) (Hrsg.): Proceedings 7th Zittau Fuzzy Colloquium. Zittau, 1999, S. 2 – 13.
 - [130] Piersma, N.; Jonker, J.-J.: Determining the Direct Mailing Frequency with Dynamic Stochastic Programming. Econometric Institute Report EI 2000-34/A, Erasums-Universität Rotterdam, 2000.
 - [131] Piersma, N.; Jonker, J.-J.; Van Den Poel, D.: Joint Optimization of Customer Segmentation and Marketing Policy to Maximize Long-Term Profitability. Econometric Institute Report EI 2002-18, Erasums-Universität Rotterdam, 2002.
 - [132] Prékopa, A.: Stochastic Programming. Kluwer Academic Publishers, Dordrecht, Boston, London, 1995.
 - [133] Press, W. H.; Teukolsky, S. A.; Vetterlin, W. T.; Flannery, B. P.: Numerical Recipes in C. Cambridge University Press, 2. Aufl., 1992.
 - [134] Puterman, M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York, 1994.
 - [135] Recktenwald, H. C.: Wörterbuch der Wirtschaft. Alfred Kröner Verlag, Stuttgart, 1990.
 - [136] Rescher, N.: Many-Valued Logic. First published by McGraw-Hill, 1969. Reprinted by Gregg Revivals, Aldershot, England, 1993.
 - [137] Ribeiro, R.A., Zimmermann, H.-J., Yager, R.R., Kacprzyk, J. (Hrsg.): Soft Computing in Financial Engineering. Physica-Verlag, Heidelberg, New York, 1999.
 - [138] Rommelfanger, H.: Fuzzy Decision Support-Systeme – Entscheiden bei Unschärfe. Springer-Verlag, Berlin u. a., 2. Aufl., 1994.
 - [139] Römisch, W.: Stability of Stochastic Programming Problems, in: Ruszczynski, A.; Shapiro, A. (Hrsg.): Stochastic Programming. Elsevier, Amsterdam, 2003, 483-554.
-

-
- [140] Ross, S.: Introduction to Stochastic Dynamic Programming. Academic Press, New York, u. a., 1983.
- [141] Scharf, A.; Schubert, B.: Marketing: Einführung in Theorie und Praxis. Schäffer-Poeschel Verlag, Stuttgart, 1994.
- [142] Schildt, H.: C++: The Complete Reference. Osborne McGraw Hill, Berkeley u. a., 3. Aufl., 1998.
- [143] Schüring, H.: Database Marketing: Einsatz von Datenbanken für Direktmarketing, Verkauf und Werbung. Verlag Moderne Industrie, Landsberg/Lech, 1991.
- [144] Segeth, W.: Elementare Logik. Deutscher Verlag der Wissenschaften, Berlin, 8. Aufl., 1973.
- [145] Seising, R. (Hrsg.): Fuzzy-Theorie und Stochastik: Modelle und Anwendungen in der Diskussion. Vieweg, Braunschweig, 1999.
- [146] Seyfarth, H.: Erfolgsorientierte Akquiseplanung. Technik & Gesellschaft, Bd. 2, 1998, S. 36.
- [147] Seyfarth, H.: Konzept einer erfolgsorientierten Akquiseplanung. Vortrag vor der VDI-Gruppe Frankfurt (Oder), IHK Frankfurt (Oder), 27.04.1998.
- [148] Strassacker, G.: Rotation, Divergenz und das Drumherum – eine Einführung in die elektromagnetische Feldtheorie. Teubner-Verlag, Stuttgart, 1986.
- [149] Sustal, J.: On the Change of the Distribution Shape of Randomized Fuzzy Variables by Filtering over Compatibility Degrees. Proceeding of the 7th Zittau Fuzzy Colloquium, 08.-10.08.1999, HTWS Zittau/Görlitz (FH), Institut für Prozesstechnik, Prozessautomatisierung und Messtechnik, 1999.
- [150] Sustal, J.: Brief vom 07.10.2002 an Klaus Weber.
- [151] Swoboda, U.: Erfolgsfaktoren der Konsumentenbanken. Finanzierung, Leasing, Factoring, Nr. 3, 1996, S. 89 – 91.
- [152] van Heesch, D.: User Manual for Doxygen 1.3.3. <http://www.doxygen.org>, 2003.
- [153] Verwilghen, N. S.: Kundensegmentierung, Risikodialog und Risikomanagement für gehobene Privatkunden: Eine Betrachtung aus finanzmarktökonomischer Sicht. Dissertation, Universität St. Gallen, 1997.
- [154] Viertl, R.: Statistical Methods for Non-Precise Data. CRC Press, Boca Racon u. a., 1995.
- [155] Viertl, R.: Einführung in die Stochastik mit Elementen der Bayes-Statistik und Ansätzen für die Analyse unscharfer Daten. Springer-Verlag, Wien, New York, 2. Aufl., 1997.
- [156] Viertl, R.: Fuzzy Daten und Stochastik. In [145], S. 244-250.
- [157] Weber, R.: Verborgene Potentiale aufdecken. Bank Magazin, Nr. 9, 1994, S. 44-46.
- [158] Weber, K.: Fuzzy Rules Generation and Fuzzy Inference Based on Frequency Distribution. In: Brewka, G. u. a. (Hrsg.): Fuzzy-Neuro Systems '99. (6th International Workshop Fuzzy-Neuro Systems '99, 18.-19.03.1999, Leipzig) Leipziger Universitätsverlag, 1999.
-

-
- [159] Weber, K.: Frequency Based Fuzzy Rules Generation and Application. In: Ochoa, A.; Soto, R.; Santana, R. (Hrsg.): Proceedings of the 2nd Symposium on Artificial Intelligence Adaptive Systems (ISAS '99), 22.-26.03.1999, Havanna. Editorial Academia, La Habana, Cuba, 1999.
- [160] Weber, K.: Fuzzy Rules Generation and Application Based on Frequency Distributions. In: Proceedings of the International Congress on Computational Intelligence: Methods and Applications (CIMA '99), 22.-25.06.1999, Rochester, New York. ICSC Academic Press, 1999.
- [161] Weber, K.; Zhaohao, S.: Fuzzy Stochastic Dynamic Programming for Process of Acquiring Customers. In: Institut für Prozesstechnik, Prozessautomatisierung und Messtechnik (Hrsg.): Proceedings 7th Zittau Fuzzy Colloquium, 08.-10.09.1999, Zittau, 1999.
- [162] Weber, K.; Würll, S.: Stochastic Multistage Process Simulation for Decision Support in Marketing. In: Proceedings of 16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation, 21.-25.08.2000, Lausanne, 2000.
- [163] Weber, K.; Zhaohao, S.: Fuzzy Stochastic Multistage Decision Process with Implicitly Given Termination Time. In: Hampel, R.; Wagenknecht, M.; Chaker, N. (Hrsg.): Fuzzy Control – Theory and Practice. Physica-Verlag, Heidelberg, 2000, S. 318 – 328.
- [164] Weber, K.; Sun, Z.: Fuzzy Stochastic Dynamic Programming for Marketing Decision Support. In: International Journal of Intelligent Systems. Bd. 15, 2000, Nr. 8, S. 763 – 784.
- [165] Weber, K.: Fuzzy Rules Generation from Data through Fuzzy Evaluation of Fuzzy Rules. In: Proceedings of 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, 30.08.-01.09.2000, Brighton. IEEE, Piscataway, New Jersey, 2000.
- [166] Weber, K., Sun, J., Sun, Z., Kliwer, G., Grothklags, S., Jung, N.: Systems Integration for Revenue Creating Control Processes. In: Journal of Revenue and Pricing Management. Bd. 2, 2003, Nr. 2, S. 120 – 137.
- [167] Weissinger, J.: Zur Theorie und Anwendung des Iterationsverfahrens. In: Mathematische Nachrichten. Bd. 8, 1952, S. 193-212.
- [168] Wets, J.-B.: Stochastic Programming Models: Wait-and-See versus Here-and-Now. In: Proceedings of IMA Conference on Decision Making under Uncertainty, 2000.
- [169] Wirth, N.: Algorithmen und Datenstrukturen. Teubner-Verlag, Stuttgart, 3. Aufl., 1983.
- [170] Woodall, W. H.: An Overview of Comparisons Between Fuzzy and Statistical Methods. Im November 1997 zugesandte unveröffentlichte Arbeit. William H. Woodall, Department of Management Science and Statistics, University of Alabama, Tuscaloosa, Alabama, E-Mail: wwoodall@alston.cba.ua.edu.
- [171] Wrobel, S.: Data Mining und Wissensentdeckung in Datenbanken. In: KI – Künstliche Intelligenz. 1998, Nr. 1. S. 6 – 10.
- [172] Würll, S.; Weber, K.: Optimal Management of Acquiring Customers by Means of Fuzzy Linear Programming. In: Proceedings 16th IMACS World Congress 2000 on
-

Scientific Computation, Applied Mathematics and Simulation, 21.-25.08.2000, Lausanne, 2000.

- [173] Yager, R. R. u. a.: Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh. Wiley, New York u. a., 1987.
- [174] Yasunobu, S.; Miyamoto, S.: Automatic Train Operation System by Predictive Fuzzy Control. In: Sugeno, M. (Hrsg.): Industrial Applications of Fuzzy Control. North-Holland, Amsterdam u. a., 1985.
- [175] Yoshida, Y. (Hrsg.): Dynamical Aspects in Fuzzy Decision Making. Physica-Verlag, Heidelberg, New York, 2001.
- [176] Zadeh, L. A.; Eaton, J. H.: An Alternation Principle for Optimal Control. In: Automation and Remote Control. Bd. 24, 1963, Nr. 3.
- [177] Zadeh, L. A.: Fuzzy Sets. In: Information and Control. Bd. 8, 1965, S. 338 – 353.
- [178] Zadeh, L. A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. In: IEEE Transactions on Systems, Man, and Cybernetics. Bd. SMC-3 1973, Nr. 1, S. 28 – 44.
- [179] Zadeh, L. A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning. Part 1. In: Information Sciences. Bd. 8, 1975, S. 199 – 249. Nachdruck in: [173], S. 219 – 269.
- [180] Zadeh, L. A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning. Part 2. In: Information Sciences. Bd. 8, 1975, S. 301 – 357. Nachdruck in: [173], S. 271 – 327.
- [181] Zadeh, L. A.: Fuzzy Sets. In: Belzer, J.; Holzman, A.; Kent, A. (Hrsg.): Encyclopedia of Computer Science and Technology. Marcel Decker, New York, 1977.
- [182] Zadeh, L. A.: The Birth and Evolution of Fuzzy Logic. In: Japanese Journal of Fuzzy Theory and Systems. Bd. 2, 1990, Nr. 2, 235 – 243.
- [183] Zadeh, L. A.: Soft Computing and Fuzzy Logic. In: IEEE Software. Bd. 11, 1994, Nr. 6, S. 48 – 56.
- [184] Zadeh, L. A.: Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive. In: Technometrics. Bd. 37, 1995, Nr. 3, 271 – 276.
- [185] Zadeh, L. A.: Fuzzy Logic, Neural Networks and Soft Computing. In: Journal A: Benelux Quarterly Journal on Automatic Control. Bd. 36, 1995, Nr. 2, S. 21 – 28.
- [186] Zadeh, L. A.: Fuzzy Logic = Computing with Words. In: IEEE Transactions on Fuzzy Systems. Bd. 4, 1996, Nr. 2, 103 – 111.
- [187] Zadeh, L. A.: What is Soft Computing? Soft Computing, Bd. 1, 1997, Nr. 1, S. 1.
- [188] Zadeh, L. A.: Some Reflections on Soft Computing, Granular Computing and their Roles in the Conception, Design and Utilization of Information/Intelligent Systems. In: Soft Computing. Bd. 2, 1998, Nr. 1, S. 23 – 25.
- [189] Zadeh, L. A.: Fuzzy Logic and the Calculi of Fuzzy Rules, Fuzzy Graphs, and Fuzzy Probabilities. In: Computers and Mathematics with Applications. Bd. 37, 1999, S. 35.

-
- [190] Zadeh, L. A.: Fuzzy Sets and Fuzzy Logic: Discovery and Controversy. Kurzfassung des Vortrags am 27.10.1999 in der Seminarankündigung, Berkeley Initiative in Soft Computing (BISC), Computer Science Division, Department of EECS, University of California, Berkeley, <http://www.cs.berkeley.edu/projects/Bisc/bisc.welcome.html>.
 - [191] Zadeh, L.A.: From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions. In: Behnam, A. u. a. (Hrsg.): Intelligent Systems and Soft Computing. Springer, Berlin u. a., 2000, S. 3 – 40.
 - [192] Zawirski, Z.: Über das Verhältnis der mehrwertigen Logik zur Wahrscheinlichkeitsrechnung. In: *Studia philosophica*. Bd. 1, 1935, S. 408 – 442
 - [193] Zimmermann, H.-J.: Fuzzy Sets, Decision Making, and Expert Systems. Kluwer Academic Publishers, Boston, Dordrecht, Lancaster, 1986.
 - [194] Zimmermann, H.-J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Boston, Dordrecht, London, 2. Aufl., 1991.
 - [195] Zimmermann, H.-J.: Unsicherheitsmodellierung. Vortrag im Mathematischen Kolloquium an der Brandenburgischen Technischen Universität Cottbus, 29.05.1999.